#AP/2113

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	09/579,670
Filing Date	May 26, 2000
First Named Inventor	Miles Aram DEFOREST
Group Art Unit	2113
Examiner Name	**Michael C. Maskulinski**
Attorney Docket Number	EMCCOR P11AUS

Total No. of Pages in this Submission: 290

ENCLOSURES *(check all that apply)*

- ■ Fee Transmittal Form
  - ■ Fee attached
- ☐ Amendment/Response
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Response to Missing Part/s Incomplete Application
  - ☐ Response to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Assignment papers *(for an Application)*
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition Routing Slip (PTO/SB/69) and Accompanying Petition
- ☐ To Convert a Provisional Petition
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Small Entity Statement
- ☐ Request for Refund
- ☐ After Allowance Communication to Group
- ☐ Appeal Communication to Board of Appeals and Interferences
- ■ Appeal Communication to Group *(Appeal Notice, Brief, Reply Brief)*
- ☐ Proprietary Information
- ☐ Status Letter
- ■ Additional Enclosure(s) *(please identify below)*:

Postcard
Transmittal of Appeal Brief - 3pgs

REMARKS

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual Name	Gary D. Clapp DAVIS & BUJOLD, P.L.L.C. — Reg. No. 29,055 **CUSTOMER NO. 020210**
Signature	
Date	June 8, 2005

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on June 8, 2005.

Type or printed name	Gary D. Clapp
Signature	Date: June 8, 2005

PTO/SB/17 (12-98)
Approved for use through 9/30/2000. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

OIPE JC145 JUN 1 0 2005 PATENT & TRADEMARK OFFICE

FEE TRANSMITTAL for FY 2005

Effective 10/01/2003. Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT: $

Complete if Known

Application No.	09/579,670
Filing Date	May 26, 2000
First Named Inventor	Miles Aram DEFOREST
Examiner Name	Michael C. Maskulinski
Group Art Unit	2113
Attorney Docket No.	EMCCOR P11AUS

METHOD OF PAYMENT (check all that apply)

■ Check ☐ Credit card ☐ Money Order ☐ Other ☐ None

■ Deposit Account:

Deposit Account Number: 04-0213

Deposit Account Name: DAVIS & BUJOLD, P.L.L.C.

The Director is authorized to: *(check all that apply)*
☐ Charge fee(s) indicated below ■ Credit any overpayments
☐ Charge any additional fee(s) during the pendency of this application
☐ Charge fee(s) indicated below, except for the filing fee to the above-identified account.

FEE CALCULATION

1. FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1011	1000	2011	500	Utility filing fee	
1012	430	2012	215	Design filing fee	
1013	660	2013	330	Plant filing fee	
1014	1400	2014	700	Reissue filing fee	
1005	200	2005	100	Provisional filing fee	____
				SUBTOTAL (1)	$

2. CLAIMS

		Extra		Fee From Below		Fee Paid
Total Claims	-20*		=	$ 50 ($ 25)	x	=
Ind. Claims	- 3		=	$200 ($100)	x	=
Multiple Dependent			=	$360 ($180)	x	=

** or number previously paid, if greater; For Reissues, see below

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	50	2202	25	Claims in excess of 20
1201	200	2201	100	Independent claims in excess of 3
1203	360	2203	180	Multiple dependent claim
1204	200	2204	100	**Reissue independent claims over original patent
1205	50	2205	25	**Reissue claims in excess of 20 and over original patent
				SUBTOTAL (2) $

***or number previously paid, if greater; For Reissues, see above*

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge-late filing fee/oath	
1052	50	2052	25	Surcharge-late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for re-examination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	120	2251	60	Ext.for reply w/in 1 mon	
1252	450	2252	225	Ext.for reply w/in 2 mon	
1253	1,020	2253	510	Ext.for reply w/in 3 mon	
1254	1,590	2254	795	Ext.for reply w/in 4 mon	
1255	2,160	2255	1,080	Ext.for reply w/in 5 mon	
1401	500	2401	250	Notice of Appeal	
1402	500	2402	250	Filing a Brief in support of an appeal	$500
1403	1,000	2403	500	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	500	2452	250	Petition to revive - unavoidable	
1453	1,500	2453	750	Petition to revive - unintentional	
1501	1,400	2501	700	Utility issue fee (for reissue)	
1502	800	2502	400	Design issue fee	
1503	1,100	2503	550	Plant issue fee	
1807	50	1807	50	Petition related to provisional applns.	
1806	180	1806	180	Submission of Info.Disclo.Stmt.	
8021	40	8021	40	Recording ea. patent assignment per property (times No.of properties)	
1809	790	2809	395	Filing a submission after final rejection (37 CFR 1.129(a))	
1810	790	2810	395	For ea.additional invention to be examined (37 CFR 1.129(b))	
1801	790	2801	395	Request for Cont.Exam.(RCE)	
1802	900	1802	900	Request for expedited examination of a design appln	

Other fee (specify)
*Reduced by Basic Filing Fee Paid SUBTOTAL (3)$500

SUBMITTED BY

Completed *(if applicable)*

Typed or Printed Name	Gary D. Clapp CUSTOMER NUMBER: 020210	Registration No.	29,055	Telephone (603) 624-9220
		Deposit Acct. No.	04-0213	Fax: (603) 624-9229
Signature	[signature]			Date: June 8, 2005

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

OIPE JC145 JUN 1 0 2005 PATENT & TRADEMARK OFFICE

In re Application of	:	Miles Aram DEFOREST and Eric Trounson MACHARDY, Jr.
Serial no.	:	09/579,670
Filed	:	May 26, 2000
For	:	FAULT TOLERANT SYSTEM SHARED SYSTEM RESOURCE WITH STATE MACHINE LOGGING
Group Art Unit	:	2113
Examiner	:	Michael C. Maskulinski
Docket	:	EMCCOR P11AUS

The Commissioner for Patents
U.S. Patent & Trademark Office
P. O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF
(PATENT APPLICATION -- 37 C.F.R. § 1.192

1. Transmitted herewith, in triplicate, is the APPEAL BRIEF in this application, with respect to the Notice of Appeal filed on April 12, 2005.

NOTE: *"Appellant must, within two months from the date of the notice of appeal under § 1.191 or within the time allowed for reply to the action from which the appeal was taken, if such time is later, file a brief in triplicate. . ." 37 C.F.R. § 1.192(a) (emphasis added).*

2. STATUS OF APPLICANT

This application is on behalf of

- ■ other than a small entity.
- □ a small entity.

A statement:

- □ is attached.
- □ was already filed.

CERTIFICATION UNDER 37 C.F.R. §§ 1.8(a) and 1.10*
(When using Express Mail, the Express Mail label number is **mandatory**;
Express Mail certification is optional.)

I hereby certify that, on the date shown below, this correspondence is being:

MAILING

■ deposited with the United States Postal Service in an envelope addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231

37 C.F.R. § 1.8(a)

■ with sufficient postage as first class mail.

37 C.F.R. § 1.10*

□ as "Express Mail Post Office to Addressee"

Mailing Label No. __________ (mandatory)

Signature

Gary B. Clapp
(type or print name of person certifying)

Date: 6/8/05

3. FEE FOR FILING APPEAL BRIEF

Pursuant to 37 C.F.R. § 1.17(c), the fee for filing the Appeal Brief is:

☐	Small entity	$250.00
■	other than small entity	$500.00

APPEAL BRIEF FEE DUE $500

4. EXTENSION OF TERM

NOTE: 37 C.F.R. § 1.704(b) ".....an applicant shall be deemed to have failed to engage in reasonable efforts to conclude processing or examination of an application for the cumulative total of any periods of time in excess of three months that are taken to reply to any notice or action by the Office making any rejection, objection, argument, or other request, measuring such three-month period from the date the notice of action was mailed or given to the applicant, in which case the period of adjustment set forth in § 1.703 shall be reduced by the number of days, if any, beginning on the day after the date that is three months after the date of mailing or transmission of the office communication notifying the applicant of the rejection, objection, argument, or other request and ending on the date the reply was filed. The period, or shortened statutory period, for reply that is set in the Office action or notice has no effect on the three-month period set forth in this paragraph."

NOTE: The time periods set forth in 37 C.F.R. § 1.192(a) are subject to the provision of § 1.136 for patent application. 37 C.F.R. § 1.19(d). See also Notice of November 5, 1985 (1060 O.G. 27).I

NOTE: As the two-month period set in § 1/192(a) for filing an appeal brief is not subject to the six-month maximum period specified in 35 U.S.C. § 133, the period for filing an appeal brief may be extended up to seven months. 62 Fed. Reg. 53,156; 1203 O.G. 63, at 84 (Oct. 10, 1997).

The proceedings herein are for a patent application and the provisions of 37 C.F.R. § 1.136 apply.

complete (a) or (b), as applicable)

(a) ☐ Applicant petitions for an extension of time under 37 C.F.R. § 1.136 (fees: 37 C.F.R. § 1.17(a)(1)-(5)) for th total number of months checked below:

	Extension <u>(months)</u>	Fee for other than <u>Small Entity</u>	Fee for <u>Small Entity</u>
☐	one month	$ 120.00	$ 60.00
☐	two months	$ 440.00	$ 225.00
☐	three months	$1,020.00	$ 510.00
☐	four months	$1,590.00	$ 795.00
☐	five months	$2,160.00	$1,080.00

FEE: $

If an additional extension of time is required, please consider this a petition therefor.

(check and complete the next item, if applicable)

☐ An extension for ______ months has already been secured, and the fee paid therefor of $___________ is deducted from the total fee due for the total months of extension now requested.

EXTENSION FEE DUE WITH THIS REQUEST $

OR

(b) ■ Applicant believes that no extension of term is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

5. TOTAL FEE DUE

The total fee due is:

Appeal brief fee $500
Extension fee (if any) $

TOTAL FEE DUE $500

6. FEE PAYMENT

■ Attached is a ■ check □ money order in the amount of $500

■ Authorization is hereby made to change the amount of

■ to Deposit Account No. 04-0213

□ to Credit card as shown on the attached credit card information authorization form PTO-2038.

WARNING: Credit card information should not be included on this form as it may become public.

■ Charge any additional fees required by this paper or credit any overpayment in the manner authorized above.

A duplicate of this paper is attached.

7. FEE DEFICIENCY

NOTE: If there is a fee deficiency and there is no authorization to charge an account, additional fees are necessary to cover the additional time consumed in making up the original deficiency. If the maximum six-month period has expired before the deficiency is noted and corrected, the application is held abandoned. In those instances where authorization to charge is included, processing delays are encountered in returning the papers to the PTO Finance Branch in order to apply these changes prior to action on the cases. Authorization to change the deposit account for nay fee deficiency should be checked. See the Notice of April 7, 1986, 1065 O.G. 31-33.

■ Of any additional extension and/or fee is required,

AND/OR

■ If any additional fee for claims is required, charge:

■ Deposit Account No. 04-0213

□ Credit card as shown on the attached credit card information authorization form PTO-2038.

WARNING: Credit card information should not be included on this form as it may become public.

DATE: 6/8/05

SIGNATURE OF PRACTITIONER

Gary D. Clapp
(Type or print name of practitioner)

REG. NO.: 29,055

Customer No. 020210

DAVIS & BUJOLD, P.L.L.C.
FOURTH FLOOR
500 NORTH COMMERCIAL STREET
MANCHESTER, NH 03101-1151

OIPE JC57 JUN 10 2005 PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF APPEALS AND INTERFERENCES

****

Appeal Number

In re Application of	:	Miles Aram DEFOREST and Eric Trounson MACHARDY, Jr.
Serial no.	:	09/579,670
Filed	:	May 26, 2000
For	:	FAULT TOLERANT SYSTEM SHARED SYSTEM RESOURCE WITH STATE MACHINE LOGGING
Group Art Unit	:	2113
Examiner	:	Michael C. Maskulinski
Docket	:	EMCCOR P11AUS

****

APPELLANT'S BRIEF

06/13/2005 SLUANG1 00000010 09579670
01 FC:1402
500.00 OP

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	:	Miles Aram DEFOREST and Eric Trounson MACHARDY, Jr.
Serial no.	:	09/579,670
Filed	:	May 26, 2000
For	:	FAULT TOLERANT SYSTEM SHARED SYSTEM RESOURCE WITH STATE MACHINE LOGGING
Group Art Unit	:	2113
Examiner	:	Michael C. Maskulinski
Docket	:	EMCCOR P11AUS

MAIL STOP APPEAL BRIEF - PATENTS
The Hon Commissioner of Patents and Trademarks
Washington, D.C. 20231

Dear Sir:

This Appeal Brief is being filed in support of Appellant's Notice of Appeal mailed on April 12, 2005, as a result of the Advisory Action issued March 30, 2005 by the Primary Examiner with regard to the Response After Final Rejection Pursuant to 37CFR 1.116 mailed on March 7, 2005 in response to the Final Rejection issued by the Primary Examiner on January 18, 2005.

I. REAL PARTY IN INTEREST: The real party in interest is:

EMC Corporation.

II. RELATED APPEALS AND INTERFERENCES: There are no related appeals or interferences in respect of the instant or any related patent application.

III. STATUS OF CLAIMS:

A. Total Number Of Claims In Application: 16

Claims in the Application are: 1 through 16

B. Status Of All The Claims:

1. Claims Canceled: None

2. Claims Withdrawn From Consideration But Not Canceled: None

3. Claims Pending: 1 through 16

4. Claims Allowed: None

5. Claims Rejected: 1 through 16

C. Claims On Appeal: 1 through 16

IV. STATUS OF AMENDMENTS

This Appeal Brief is being filed in support of Appellant's Notice of Appeal mailed on April 12, 2005, as a result of the Advisory Action issued March 30, 2005 by the Primary Examiner with regard to the Response After Final Rejection Pursuant to 37CFR 1.116 of March 7, 2005. In the Response After Final Rejection of March 7, 2005 the Applicant submitted arguments rebutting the grounds for rejection expressed in a Final Office Action of January 18, 2005 and proposed amendments to claims 1, 3, 5, 7, 9, 11, 13 and 15 addressing issues raised by the Examiner in the Final Rejection as regards support in the claims for the arguments rebutting the grounds for rejection. The Examiner declined to enter either the arguments or the claim amendments submitted in the Response After Final Rejection Pursuant to 37CFR 1.116 of March 7, 2005 on the grounds that the proposed amendments would raise new issues requiring further consideration and/or search. No further amendment has been filed subsequent to the response of June 14, 2004, which included both arguments distinguished the invention of the prior art and an amendment to claim 11.

The accompanying Appendix A, Pending Claims, thereby contains claims 1-16 as pending after the Final Rejection of January 18, 2005. In addition, the accompanying Appendix B, Proposed Amended Claims, contains claims 1-16 with the claim amendments proposed in the Response After Final Rejection of March 7, 2005.

V. SUMMARY OF THE INVENTION

The present invention is directed to system resource for performing system resource operations requested by a client of the system resource, such as a file server performing file read and write transactions requested by a client.

The system resource includes a system resource sub-system and a control/processing sub-system wherein the storage sub-system may be, for example, a file system, and the control/processing sub-system may include a resource control processor, such as a file system processor, performing system resource operations in response to client requests and controlling operations of the system resource sub-system, such as file reads and writes.

According to the present invention, the system resource further includes a state machine logging mechanism for extracting, storing and restoring state machine information representing the state of operation of the system resource and of the system resource operations, such as file transactions. The state machine logging mechanism includes a state machine log generator for extracting state machine information defining a current state machine representing a current state of execution of a system resource operation and a state machine log mechanism for storing a sequence of one or more state machines representing the sequential, detailed, state machine level operations of the system in executing the requested transactions. The state machine log generator is responsive to the restoration of operation of the system resource after a failure of the system resource for reading the state machine information from the state machine log mechanism and restoring the state of execution of a current system resource operation.

Also according to the present invention, the state machine log mechanism includes a state machine log mirroring mechanism operating separately from the control/processing sub-system and communicating with the state machine log generator through a local high speed data link for receiving and storing mirror copies of the state machine information. The state machine log mirroring mechanism may then respond to a failure of the control/processing subsystem and to a

subsequent restoration of operation of the file server after a failure of file server operations to read the state machine information from the state machine log mirroring mechanism and to restore the state of execution of a file transaction.

In this regard, the state machine log mirroring mechanism operates separately from but concurrently with and in cooperation with the control/processing sub-system in such a manner that the state machine log mirroring mechanism is capable of remaining in operation even upon failure of the control/processing sub-system. That is, the control/processing subsystem and state machine log mirroring mechanism operate concurrently and cooperatively but are independent from one another in such a manner, for example, by being powered from separate power sources, that a failure in the control/processing subsystem will not in itself result in a failure of the state machine log mirroring mechanism.

A comprehension and understanding of the terms “state machines”, “machine state”, and “state machine information” is necessary to a comprehension and understanding of the fundamental distinctions of the present invention over the cited prior art. For these reasons, therefore, and as described and defined in the specification and claims, a "state machine" is a representation or model of a machine or system, such as a system resource, a file server, a sub-system or logical or functional element of a system or sub-system of any form, that executes operations as a sequence of discrete "machine states", also referred to as “states”. Each “machine state”, or “state” of a sequence of machine states is defined during or at an interval or point in time during in the machine state is not changing, which is typically and for example during the interval between clock pulses. The present state and next operating states of a state machine are described and defined by and are comprised of the current state of the machine, the state functions of the machine itself, that is, the logic and circuit functions implemented in the machine that determine the responses or changes in state of the machine as a result of a current operating state

and any current control or data inputs that effect the current state, and any control or data inputs that will effect the present or next state of the state machine.

In summary, therefore, a "state machine" is therefore comprised of and is defined and described as a sequence of one or more state machines wherein each state machine in the sequence of state machines is defined by the current state at that point in time. That is, each state machine in a sequence of one or more state machines is defined and described by the control and data values residing in the machine, and the state functions of the machine, that is, the functions or operations that will be executed by the state machine to result in the next state machine, and any current control or data inputs that will operate or be used in defining the next state machine in the sequence.

The state functions of a system, that is, the responses or changes in state of the machine as a result of a current operating state and any current control or data inputs that effect the current or next states of the system, are determined by the logic and circuit functions implemented in the system determine. As such, the state functions of a given system for all possible states of the system are typically fixed and are thereby explicitly or implicitly known and need not be specified for each state individually. As a consequence, the state machine of a system at any point in the operations performed by the system, that is, the current state of the system and, as a consequence, the next state of the system, is fully defined and described by the machine state at that point, that is, by the control and data values residing in the state machine or present as inputs to the state machine at that point. As such, it is not necessary to define the logic functions of the system individually for each possible state as the logic functions are known and invariable and only the state of the system varies. A sequence of operations executed by the system may therefore be defined and described by a corresponding sequence of machine states.

It must therefore be emphasized that the state machine logging mechanism of the present invention extracts and stores "state machine information" defining one or more sequential "state machines" during the execution a resource operation or transaction rather than storing information pertaining to the system operation only at the start or conclusion of an operation or transaction, as is typical in transaction logging type systems.

By way of illustration, conventional transaction logging systems of the prior art typically record the operations of a system by recording the parameters of an operation to be performed at the start of the operation and possibly at the end of the operation and the recorded "parameters" typically include only the command, request or instruction initiating the operation and the input data provided to the operation or the output data resulting from the operation. A conventional logging operation for a system operation may thereby be thought of as a system capturing a "snap-shot" of the minimum information required to define an operation, that is, the instruction, request or command initiating the operation and the data operated upon in the operation.

It is well understood by those of ordinary skill in the relevant arts, however, that any operation in a system, such as a file transfer, a file read or a file write, is comprised of a sequence of sub-operations wherein, as recognized in the state machine mechanisms of the present invention, each sub-operation is a complete operation in itself and essentially involves a single operation performed on a single body of data. A given operation or a completed portion of an interrupted or aborted operation can therefore be accurately and completely restored from a log record only by accurately and completely recording the sub-operations of the sequence of sub-operations comprising the operation, which is in accordance with the prior art limitation that only fully completed operations are logged and can be restored.

The present invention, however, provides a mechanism for logging and mirroring each of the sub-operations of the sequence of sub-operations comprising a given operation of the system

by recording each sub-operation as a state machine, that is, as the machine state of a state machine, of a sequence of state machines representing or modeling the operation of the system.

Stated simply, therefore, the state machine logging mechanism of the present invention thereby provides greater "granularity" in recording the execution of an operation or transaction in capturing and storing of a sequence of state machines reflecting the sequential sub-operations of the system during the execution of a requested transaction, thereby recording and representing each transaction or operation as a sequence of state machines. Because a state logging and mirroring mechanism of the present invention will record each of the system sub-operations that comprised a system operation or transaction as an individual state machine of a sequence of state machines, a state logging and mirroring mechanism of the present invention can record events within the execution of any system operation or transaction. This greater granularity in turn allows the restoration of transactions and operations at the point at which they were interrupted, that is, during the execution of any of the sub-operations comprising a transaction or operation, rather than at only a single point at the beginning or end of the transaction or operation.

The state machine logging mechanism of the present invention thereby allows the restoration of a resource server or transaction at a point during the performance or execution of the operation or transaction. The state machine logging mechanism of the present invention thereby also provides greater assurance that a given resource server operation or transaction is captured for subsequent restoration, if needed, because the necessary information is captured at each stage in the execution of the transaction as a sequence of points in the execution of the operation or transaction, rather than only when the operation or transaction is completed. Stated another way, each change in the system during the execution of a transaction defines a new state of the machine, which is captured and stored as the next state machine in the sequence of state machines defining the transaction. As a consequence, no essential step during the execution of

a transaction is or can be lost, thereby allowing reliable reconstruction or restoration of any transaction.

Lastly, it must be noted that the invention as described in the specification and as recited in certain of the claims is directed to an implementation of the present invention in a system resource having a single control/processing sub-system having an associated state machine log generator and a state machine log and is recited as such in claims 1, 3, 5, 7, 9, 11, 13 and 15.

The present invention as recited in the claims is, however, also directed to an implementation of the present invention wherein the state machine logging mechanism further includes a state machine log mirroring mechanism that is operationally separate and independent form the control/processing sub-system, such as in claims 2, 4, 6, 8, 10, 12 14 and 16. That is, in the recitations of claims 2, 4, 6,8, 10, 12, 14 and 16, the state machine logging mechanism has a state machine log situated in or in association with the control/processing sub-system and a separate state machine log mirroring mechanism that is effectively external and independent of the control/processing sub-system. In this regard, it must be recognized that the state machine log mirroring mechanism of a given control/processing sub-system is not independent from the control/processing sub-system or the state machine log of that control/processing sub-system as regards the state machine log mirroring functions, but is separate and independent from that control/processing sub-system as regards basic support functions, such as power to the separate state machine log mirroring mechanism. As such, and for this reason, the separate state machine log mirroring mechanism can continue operation if there is a failure of the control/processing sub-system.

Further in this regard, it must be noted that the state machine log mirroring mechanism is not a separate system analogous to the control/processing sub-system and is not a complete, self-contained and full function system in the manner of a general purpose computer. Instead, the state

machine log mirroring mechanism is a specific purpose mechanism designed and constructed only to perform the state machine log mirroring function of capturing and storing a copy of the state machine information and state machines that are stored in the state machine log associate with the state machine log generator that is associated directly with the control/processor sub-system. As such, it will be clear to those of ordinary skill in the arts that the separate state machine log mirroring mechanism does not and cannot operate as a "back-up" unit for the control/processing sub-system in processing requests for operations as the state machine log mirroring mechanism simply does not have the functionality for this type of operation. The state machine log mirroring mechanism instead operates solely as a specific purpose mechanism designed and constructed only to perform the state machine log mirroring function of capturing and storing a copy of the state machines comprising the control/processing sub-system while it is processing transactions.

It must also be noted that the claims are also directed to implementations of the present invention having dual control/processing sub-systems with associated state machine logging mechanisms and state machine log mirroring mechanisms. In these implementations, each control/processor sub-system has an associated state machine logging mechanism for its own transactions or operations and a state machine log mirroring mechanism associated with the other control/processing sub-system. As such, each control/processing sub-system maintains its own state machine logging mechanism and maintains a state machine log mirroring mechanism for the other control/processing sub-system and processes transactions independently of the other control/processing sub-system.

In this regard, it must also be noted that in the system resources or file servers having dual control/processing sub-systems, the two control/processing sub-systems operate concurrently, independently and in parallel with each other and with each control/processing sub-system performing its own system resource operations or transactions, and with each control/processing

sub-system providing only a residence and support for the state machine log mirroring mechanism of the other control/processing sub-system.

As such, and as described in the specification and recited in the claims, neither control/processing sub-system provides “back-up” for the other control/processing system in the sense of standing by ready to perform or execute transactions or operations for the other control/processing sub-system if the other control/processing sub-system should fail. It must also be noted that because of the concurrent, independent and parallel operation of each control/processing sub-system, with each control/processing sub-system executing only the requests for transactions or operations that are directed to it. As a result, the full processing power of the two control/processing sub-systems is available at all times to process requests from clients. If one control/processing sub-system fails, the other control/processing sub-system continues operative to perform the request directed to it and to maintain the state machine information necessary to subsequently restore the failed control-processing sub-system. In addition, the operative one of the control/processing sub-systems may perform requests that would be directed to the other control/processing sub-system if those requests are re-directed to the operative control/processing sub-system. It will therefore be apparent to those of ordinary skill in the arts that only in the event of a failure in one of the control/processing sub-systems of the present invention is the total transaction processing capability of the system reduced to that levels provided by the normal operation of the “back-up” systems of the prior art, such as taught by Rastogi et al. ‘449.

To further illustrate by example, in a conventional “back-up” system of the prior art, two identical systems arranged in parallel with one unit being designated as the primary unit and the other as the “back-up” unit. The primary unit typically performs all operations requested of the system, while the “back-up” unit is typically idle until it is required to replace the other unit in

performing the operations, so that only one half of the total power and resources of the system are available at any time.

VI. ISSUES:

The issues presented for appeal are:

(a) Whether the Applicant's characterization of state machines in the presently pending claims is incorrect in view of the textbook Logic and Computer Design Fundamentals by Mano and Kime which the Examiner interprets as stating that a state machine consists of a sequence of states rather than as a sequence of state machines.

(b) Whether the claim amendments submitted by the Applicant in the Response After Final Rejection Pursuant to 37CFR 1.116 of March 7, 2005 in response to the Final Rejection issued January 18, 2005 would require further search and/or consideration.

(c) Whether claims 1, 3, 5, 9, 11, 13 and 15 are unpatentable as anticipated under 35 U.S.C. § 102 over U.S. Patent No. 6,205,449 to Rastogi et al. for a SYSTEM AND METHOD FOR PROVIDING HOT SPARE REDUNDANCY AND RECOVERY FOR A VERY LARGE DATABASE MANAGEMENT SYSTEM, hereafter referred to as "Rastogi et al. '449".

(d) Whether claims 2, 4, 6, 8, 10, 12, 14 and 16 under 35 U.S.C. § 103(a) over Rastogi et al. '449 and further in view of U.S. Patent No. 5,513,314 to Kandasamy et al. for a FAULT TOLERANT NFS SERVER SYSTEM AND MIRRORING PROTOCOL, hereafter referred to as "Kandasamy et al. '314".

VII. GROUPING OF CLAIMS:

For purposes of the present Appeal, the grouping of claims as defined by the claim rejections are:

Group I: claims 1, 3, 5, 7, 9, 11, 13 and 15 under grounds for rejection (c) wherein claims 3, 5, 7, 9, 11, 13 and 15 are believed to stand or fall with the allowability of claim 1; and

Group II: claims 2, 4, 6, 8, 10, 12, 14 and 16 under grounds for rejection (d) wherein claims 4, 6, 8, 10, 12, 14 and 16 are believed to stand or fall with the allowability of claim 2.

VIII. ARGUMENTS

(a) It is the Applicant's belief that the Examiner's holding that the Applicant's characterization of state machines in the presently pending claims is incorrect in view of the textbook Logic and Computer Design Fundamentals by Mano and Kime, which the Examiner interprets as stating that a state machine consists of a sequence of states rather than as a sequence of state machines, is in error for the following reasons.

The Examiner has not explicitly rejected any of the pending claims under 35 U.S.C. 102, 35 U.S.C. 103 or 35 U.S.C. 112 on the grounds that the Applicant's characterization of state machines in the presently pending claims is incorrect in view of the textbook Logic and Computer Design Fundamentals by Mano and Kime. The Examiner has held, however, that the Applicant's characterization of state machines is integral to the Applicant's arguments distinguishing the invention as recited in the claims over the cited prior art and that, for this reason, among others, the Examiner has found the Applicant's arguments to be not persuasive. The Examiner's holding that the Applicant's characterization of state machines is therefore material to the Examiner's rejections of the claims under grounds (c) and (d).

It is the Applicant's belief that, for the following reasons, the Applicant's characterization of state machines in the presently pending claims is correct in view of the textbook Logic and Computer Design Fundamentals by Mano and Kime for the following reasons.

First, and in general, an invention typically embodies new concepts or viewpoints not anticipated or perceived in the prior art. As such, a prior art reference, such as a textbook, can by its very nature describe and define only what is in the prior art and an invention cannot be defined and bound only to statements in textbooks or there would be few or no new inventions.

In addition, and referring to the above discussions of state machines in the context of the present invention, the specification of the present Application defines the terms "state" and "state machine" according to the present invention and the Applicant is entitled to recognition of terms so defined so long as those definitions do not explicitly mis-represent the terms as understood in the art or lead to misunderstandings due to lack of adequate definition. As discussed above, the terms "state" and "state machine" as defined and employed in the present Application are not in conflict with those terms as understood in the art but are instead developments and extensions of the meanings of those terms over what was known in the prior art and, moreover, are fully and completely described, defined and discussed in the specification of the present Application.

More specifically, the present invention defines a state machine, for purposes of the present invention, as representing the state existing in a system during one step in the execution of an operation comprised of a sequence of such steps. As discussed, this definition is not in conflict with the classical definition of a state machine as stated in, for example, <u>Logic and Computer Design Fundamentals</u>, by Mano and Kime and as summarized by the Examiner in paragraph 7 of the Final Action, but represents a different viewpoint of the relationship between state machines and operations or processes executed in a system.

In this regard, the Applicant concurs that a state machine is classically defined by the outputs resulting from each possible combination of inputs, and that the set of all outputs for all possible inputs represents the internal logic of the state machine. The present invention recognizes, however, that a "sequence of states" defining a state machine may be comprised of

a single state, that is, of one specific combination of inputs and the corresponding one specific combination of outputs, as is inherent in the description and definition of a state machine in Logic and Computer Design Fundamentals, by Mano and Kime. In a like manner, and in accordance with the description of the present invention in the specification of the Application, a state machine, such as a system resource, may be considered as comprised of a plurality of state machines, such as a storage sub-system state machine, a control/processing sub-system state machine and a file system processor state machine. Each such state machine may in turn, again in accordance with the description of the present invention as described in the specification of the Application, be defined as a sequence of single step state machines wherein a given single step state machine is defined by the state existing during a single step of a sequence of one or more steps that are executed in the execution of, for example, a file transaction operation.

It is therefore the Applicant's belief and position that the meaning of the terms "state" and "state machine" as defined and used in the specification and claims of the present invention are therefore fully defined in the specification and claims of the present Application. It is further the belief and position of the Applicant that the terms in question are not in conflict with the classical definitions of these terms as set forth in Logic and Computer Design Fundamentals, by Mano and Kime and that the definitions set forth in Logic and Computer Design Fundamentals, by Mano and Kime do not preclude the uses of the terms in the present Application.

Returning to the issue originally expressed by the Examiner, it must first be noted that at no place in the claims does the Applicant in fact recite a state machine as a sequence of state machines. The claims instead recite that an operation, such as a transaction operation, is comprised of a sequence of one or more steps and that the system of the present invention represents each transaction operation as a sequence of state machines wherein each state machine represents and is defined by the state in a single step in the sequence of steps executed

in executing a transaction operation. This is very different from reciting that a state machine is a sequence of state machines.

The Applicant therefore cannot see how the Examiner's understanding of a state machine as a sequence of state machines was derived from the claims of the present invention. The Applicant must therefore assume that the Examiner arrived at this misunderstanding from the Applicant's arguments that were presented in an attempt to explain the distinctions between the present invention and the prior art. If so, the Applicant must apologize for confusing the Examiner and hopes that the above discussions clarify what is meant by the terms in questions.

The Applicant must point out, however, that the statement that a state machine is comprised of a sequence of state machines is correct according to the present invention, and that the concept of a state machine comprised of a plurality of state machines, each of which is defined as a sequence of single step state machines, is in fact one of the concepts by which the present invention is distinguished over the prior art, including <u>Logic and Computer Design Fundamentals</u>, by Mano and Kime.

To clarify this point, if a system is defined as a state machine, then according to Mano and Kine the system state machine is defined by all possible states that could exist in the system, which in turn is represented by all possible combinations of system inputs and the corresponding system outputs. It will be apparent that all combinations of system inputs and the corresponding system outputs that can exist in the system will include all operations that can be performed by the system wherein each operation will be comprised of a sequence of states selected from all of the states that can exist in the system state machine. According to the present invention, therefore, each operation that can be executed by the system state machine may be regarded as itself comprising an operation state machine wherein an operation state machine is defined by the sequence of

system states that exist during execution of the operation and wherein the operation states and their sequence are selected from the set of all states that can exist in the system state machine.

Continuing with this concept, and according to the present invention, it will also be apparent that each operation that can be performed by the system state machine will be comprised of a sequence of steps that are executed by the system state machine in performing the operation. Each step, however, is represented by the state existing in the system during the execution of that step and, according to the present invention, is therefore represented by a state machine that is defined by the state existing during execution of that step, which may be referred to as a "step state machine". An operation performed by a system, that is, by a system state machine, is thereby represented by a sequence of "step state machines" wherein each step state machine represents the state existing in the system state machine during execution of the corresponding step of the operation. According to the present invention, therefore, an operation, which can be represented by an operation state machine, will be comprised of a sequence of steps wherein each step is defined by the state existing during execution of that step and wherein the state existing during each step defines a step state machine representing the step. The operation state machine is therefore, and according to the present invention, comprised of a sequence of step state machines.

(b) It is the Applicant's belief that the Examiner's holding that the claim amendments submitted by the Applicant in the Response After Final Rejection Pursuant to 37CFR 1.116 of March 7, 2005 in response to the Final Rejection issued January 18, 2005 would require further search and/or consideration is in error for the following reasons.

As will be apparent from the amended claims presented in Appendix B, Proposed Amended Claims, which contains claims 1-16 with the claim amendments proposed in the Response After Final Rejection of March 7, 2005, the amendments submitted in the Response After Final Rejection

of March 7, 2005 were submitted solely in a good faith effort to address the Examiner's issues regarding pertaining to the use of the terms "sequence of states" and "sequence of state machines" as discussed with regard to Issue (a) above. As such, the proposed amendments were in the nature of an amendment to meet a 35 U.S.C. 112 type of issue, and were not submitted to, and did not, alter the scope, meaning or subject matter of the claims or to define around the prior art in any way.

As such, it is the Applicant's belief and position that the amendments proposed in Response After Final Rejection of March 7, 2005 did not necessitate a new search or further consideration because the proposed amendments would not and did not alter the scope, meaning or subject matter of the claims and are not submitted to define around the prior art in any way.

In addition, it should be noted that throughout the prosecution history of the present Application, which includes a previous cycle of actions and responses, including a proceeding Final Rejection and a Request for Continued Examination, the Examiner has essentially relied solely on the two references cited in the present Final Rejection, U.S. Patent No. 6,205,449 to Rastogi et al. and U.S. Patent No. 5,513,314 to Kandasamy et al. It is extremely unlikely that the Examiner would be able to discover new pertinent prior art at this time as a result of the amendments submitted in the Response After Final of March 7, 2005, particularly when the responses to previous actions have in fact submitted substantive amendments to the claims and, for that reason, would more validly justify a new search or further consideration than do the presently proposed amendments.

(c) It is the Applicant's belief that the rejection of claims 1, 3, 5, 9, 11, 13 and 15 as unpatentable as anticipated under 35 U.S.C. § 102 over U.S. Patent No. 6,205,449 to Rastogi et al. for a SYSTEM AND METHOD FOR PROVIDING HOT SPARE REDUNDANCY AND

RECOVERY FOR A VERY LARGE DATABASE MANAGEMENT SYSTEM, hereafter referred to as "Rastogi et al. '449", is in error for the following reasons.

The following will first discuss the teachings of Rastogi et al. '449 and the relevance of Rastogi et al. '449 to the present invention as recited in claims 1, 3, 5, 7, 9, 11, 13 and 15, and will then address certain specific issues raised by the Examiner in the Office Action of January 18, 2005, the Applicant's responses to those issues having been presented in the Response of March 7, 2005.

Rastogi et al. '449 describes a system having "hot spare" support wherein the system includes two identical computer systems communicating through a network and wherein, at any given time, one of the computer systems is designated as the "primary" computer system and the other is designated as the "secondary" computer system. The computer system that is designated as the primary computer system performs all operations, that is, all transactions, of the system and generates a log of all such transactions. The computer system that is designated as the secondary computer system does not perform any operations while the primary system is in operation, but is instead a "back-up" system that stores a copy of the transaction log of the primary system. If there is a failure in the current primary computer system, the secondary computer system becomes the primary computer system and assumes execution of all transactions directed to the system, starting with the copy of the primary system transaction log stored in the secondary computer system, which then becomes the primary computer system.

In Rastogi et al. '449 the primary and essentially the sole object of storing a copy of the current primary system transaction log in the secondary system is to maintain the primary and secondary systems in "synchronization" so that the secondary computer system can assume execution of the transactions directed to the system with minimum loss of the preceding

transactions that have been executed by the computer system that was previously the primary computer system.

Rastogi et al. '449 states that, for this purpose, the primary and secondary computer systems each stores a record of the "state" of the computer systems. It will be readily recognized by those of ordinary skill in the relevant arts, however, that Rastogi et al. '449's use of the term "state" differs fundamentally from the term "state" as used in the specification and claims of the present Application.

In particular, in Rastogi et al. '449 the term "state" refers to bits of information stored in the primary and secondary systems that indicate, in each system, whether the system is the current primary system or is the current secondary system and whether the two computer systems are in "synchronization", that is, have matching copies of the primary computer system transaction log. At any given time only one computer system can be the current primary computer system and can execute the system transactions. The secondary system will always be idle as regards the execution of transactions and will function solely to store a copy of the primary computer system transaction log while waiting to assume execution of the transactions upon failure of the current primary computer system.

It will therefore be apparent to those of ordinary skill in the arts that in the teachings of Rastogi et al. '449 the term "state" has no relationship or meaning with regard to the state of operation of a system at each step in executing a transaction, or even to the actual execution of transactions, but instead relates only to the overall current functional assignments of the two systems and, in particular, to which one is assigned to execute transactions and which one is assigned to store a copy of the transaction log, and not how the transactions are executed.

As taught by Rastogi et al. '449, the primary system transmits a copy of the primary system transaction records to the secondary system through a network connection in either of two

circumstances, depending upon the specific designed operation of the Rastogi et al. '449 system. In one circumstance, the primary system transaction log is transmitted to the secondary each time the transaction log in the primary system is "flushed" to disk, that is, is moved from the primary system memory space to the primary system mass storage device for long term storage. In the second circumstance, the primary system will flush its local copy of the transaction log to its own disk when the primary system has transmitted a copy of the transaction log to the secondary system and has received an acknowledgment of the transmission from the secondary system.

It is, therefore, apparent that the present invention is distinguished over and from the Rastogi et al. '449 system for a number of fundamental reasons, which are recited in claims 1, 3, 5, 7, 9, 11, 13 and 15.

For example, and in complete and fundamental contrast from Rastogi et al. '449, the system of the present invention captures and stores state machines representing the detailed operation of the system in the execution of each transaction and stores these state machines while the corresponding transaction is being executed, so that the execution of a transaction may be continued or resumed at any time, not just at the end or beginning of a transaction.

It will therefore be apparent from the above discussions that the present invention is fundamentally distinguished over and from the system taught by Rastogi et al. '449 in essentially every respect. For example, the transaction log of the Rastogi et al. '449 system captures or generates transaction records only when the transactions are completed, that is, at the end of the execution of each transaction.

In fundamental contrast from the present invention, which captures each transaction and the sub-operations within each transaction as a sequence of state machines, Rastogi et al. '449 does not and cannot capture and preserve records of the sub-operations within and comprising a transaction or operation.

This distinction is further supported and emphasized in that the Rastogi et al. '449 system actually records or stores a record of a completed transaction only at the conclusion of the transaction, when the record is transferred into mass storage, as any record that the Rastogi et al. '449 system generates before the completion of a transaction is held only in volatile memory and is thus lost if the system fails. In a like manner, a copy of a transaction record is transmitted to and stored in the secondary computer system only when the record is transferred to the primary system mass storage, so that the back-up copy of the transaction record can also be lost if a system failure occurs before the end of the transaction.

In fundamental contrast from Rastogi et al. '449, the system of the present invention captures and records and mirrors each sub-operation in an operation as a state machine and as and when each sub-operation is executed, so that a transaction or operation is continuously recorded at each sub-operation point as the transaction or operation is executed. As a consequence, if there is a failure of a control/processor sub-system during the execution of a transaction, the sub-operations of the transaction will have been captured and recorded and mirrored up to the sub-operation in which the failure actually occurred. As such, only a part of a transaction is lost and the transaction can be restored up to the point of failure. The Rastogi et al. '449 system cannot and does not provide this level of capture and mirroring and can capture and restore only completed transactions.

In further fundamental contrast from the present invention, it must be noted that the transaction records captured by Rastogi et al. '449, that is, the data comprising a record of a completed transaction that is captured by Rastogi et al. '449, has no relationship to a state machine representing the control and data values present in a state machine system and representing the operating state of a system of operation at a given time.

To explain in greater detail, it must first be noted that Rastogi et al. '449 describes a transaction as a sequence of operations at one or more levels and describes a transaction log as storing such transactions. Rastogi et al. '449, however, and for example at column 8, lines 3-15, speaks only of recording completed transactions and not of capturing, recording or restoring the sub-operations of a transactions.

Further in this regard, it must be noted that the only mention of "state" by Rastogi et al. '449 is, for example, at column 5, lines 29, to column 7, line 15, wherein Rastogi et al. '449 defines "state" within the context and meaning of the teachings of Rastogi et al. '449 as being no more than stored information indicating which is the primary system and which is the secondary system and whether the primary and secondary systems are synchronized.

Thus, while Rastogi et al. '449 uses the word "state", it must be noted and understood that Rastogi et al. '449 defines the term "state" to have a meaning within the teachings of Rastogi et al. '449 that has no relationship to the defined meaning of the term "state" in the present Application and the claims thereof. In this regard, it must also be noted that both usages of the word "state" are correct as being within the commonly accepted meanings of the word, but that the uses have separate and different meanings that are dependent upon the contexts in which they are used. Therefore, while Rastogi et al. '449 uses the word "state", the meaning of the word "state" as employed and defined in the present Application and claims is fully and fundamentally distinguished from the meaning of the word "state" as used in Rastogi et al. '449.

Further in this regard, it must be noted that Rastogi et al. '449 essentially only describes representing a transaction in the primary system transaction log in terms of what are in fact the instructions or commands initiating each operation at each level, and perhaps any data input to the operation. At no point does Rastogi et al. '449 describe or even mention a "state machine system", a "state machine", system "state" as represented by the control and data values residing in the

system at a given point during the execution of a transaction and defining the "state machine" of the system at that point.

The only reasonable conclusion that could be reached by one or ordinary skill in the arts is therefore that Rastogi et al. '449 is, as stated, referring to a transaction log as containing only either the instructions or commands initiating each a transaction, and any input data, or the data resulting from a transaction at the completion of the transaction and that, in complete contrast from the present invention as described and claims, Rastogi et al. '449 does not mention, describe or teach in any way the use of state machines to capture, record and mirror the sub-operations of transactions as state machines.

To further illustrate this distinction by an example, it must be noted that each sub-operation or step in the execution of a transaction is often effected by the outcome of a preceding sub-operation or step in the execution of the transaction and that, in most systems, any such information from a preceding sub-operation of step will be appropriately stored in the processing unit. A conventional system such as that taught by Rastogi et al. '449, however, may well miss such information from a preceding sub-operation or step within an operation as the log stores only the current instruction or command and input data for the operation as a whole or the data resulting from completion of the operation as a whole, and does not store data or information for the sub-operations or steps within the operation.

A state machine system of the present invention, however, because the state machine log stores, at each sub-operation or step in the execution of a transaction, a state machine representing the state of the system during that sub-operation or step. As described and recited in the present Application and claims, each such state machine includes the control and data values present in the system at that time and effecting the execution of the step, including

information from a preceding sub-operation or step that may effect the execution of the current sub-operation or step.

The above discussed characteristics of the Rastogi et al. ‘449 system, that is, the logging of transactions rather than state machines and the logging of a transaction only when the primary system flushes its transaction log to local mass storage, results in yet other fundamental distinctions between the present invention and the teachings of Rastogi et al. ‘449.

For example, because the Rastogi et al. ‘449 system captures transactions, that is, the instructions and data initiating a transaction or at the completion of a transaction, rather than a sequence of state machines, the Rastogi et al. ‘449 system is essentially limited to capturing, recording and restoring each transaction as an entity. The Rastogi et al. ‘449 system therefore cannot and does not capture or record the sub-operations within a transaction and thus cannot restore a transaction any sub-operation within the execution of the transaction, but instead captures only the beginning or end of a transaction. It is for these reasons that the Rastogi et al. ‘449 system contains both a “redo” log, so that a transaction can be reexecuted from the start, and an “undo” log, so that a transaction can be “undone”, or canceled, by “undoing” the transaction from the end.

These fundamental distinctions between the present invention and the teachings of Rastogi et al. ‘449, are further supported by the fact that the Rastogi et al. ‘449 system “logs” a transaction in the back-up log in the secondary systems only when the primary system flushes its transaction log to local mass storage. In other words, the “log” of a given transaction is stored solely in the volatile memory of the primary system until the log is flushed to the primary system mass storage, and is not until the log is flushed to the primary system mass storage that the log is also “flushed” to the secondary system to be stored in the secondary system mass storage.

While Rastogi et al. '449 does not address the implications of these log or flush operations explicitly, it is clear that a "flush" of a transaction from memory to mass storage will typically and normally take place only at the conclusion of the transaction as this will thereby require the storage of only the minimum amount of information about the transaction in order to allow the transaction to be "redone" or "undone". This interpretation of the teachings of Rastogi et al. '449 are supported by Rastogi et al. '449 at, for example, column 8, lines 3 through 61, and in more detail at column 8, line 3 through column 11, line 8. It is noted, in this regard, that Rastogi et al. '449 does not in fact state that a log entry contains information taken during the execution of the transaction, but only that the stored information includes only the information essential to reconstruction of the transaction as an entity. The system description provided in Rastogi et al. '449, however, is compatible with a system that stores only the data and commands initiating a transaction or the data resulting at the completion of a transaction is completed.

It must also be noted in this regard that Rastogi et al. '449 explicitly teaches only the storage of transactions, such as at column 8, lines 3 through 61, for example, and does not discuss or even mention or suggest storing a sequence of state machines representing a transaction or any other form of sub-operation data from within the execution of a transaction. Again, Rastogi et al. '449 speaks only of "redoing" and "undoing" a "transaction" as an entity, which is defined as a set of sub-operations and which requires only either the transaction starting data and commands or the transaction results. Rastogi et al. '449 does not even mention restoring or resuming a transaction at any of the sub-operations comprising the transaction, but only of redoing or undoing a transaction in its entirety.

As such, it is clear that Rastogi et al. '449 does not and cannot capture or store state information of any sort, including state machines, state machine data or the data and control values residing in the control/processing sub-system state machine at each step in the execution of a

transaction. It is also clear to those of ordinary skill in the art that this distinction is so fundamental and so basic to the entire design and operation of the two systems as to render the teachings of Rastogi et al. '449 irrelevant with respect to the present invention.

Lastly in this regard, it is noted that the Examiner states that Rastogi et al. '449 teaches the use of state information because the secondary system is available "immediately" to take over the functions of the primary system upon failure of the primary system. It must be noted, however, that not only does Rastogi et al. '449 not even mention state, except with regard to which processor is the primary and secondary processor and whether they are synchronized, that the use of the term "immediately" as used in Rastogi et al. '449 must be read and interpreted in light of the remainder of the teachings in Rastogi et al. '449.

That is, and for example, Rastogi et al. '449 clearly teaches that the logging of a transaction in the back-up log in the secondary systems occurs only when the primary system flushes its transaction log to local mass storage, which is typically only at the end of the transaction. This thereby indirectly states that not only will an "unlogged" or "unflushed" transaction be lost if the is a power failure to the primary processor before the log is flushed, but that since the logging takes place only at a "transaction rate", the term "immediately" does not imply any great speed of response but only that it will take place at the normal paces of transaction processing. This position is further supported when it is noted that the primary system and the secondary system in which the transactions are logged are interconnected through a network so that any restoration of transaction data must take place at network transaction speeds, which is normally a rather slow "immediately". A more reasonable interpretation of the Rastogi et al. '449 teachings is that the secondary system is ready to assume the transaction processing operations of the primary system immediately upon being informed of a failure of the primary system, which is a very different matter and operation from the transaction logging operations.

Support for this position may be found in Rastogi et al. '449 at, for example, column 3, line 22, through column 4, line 59, column 5, lines 9 to 44, column 7, line 38 to column 8, line 2, column 8, lines 3-15, column 8, line 24 through column 10, line 30.

To consider this distinction further, and in contrast from the Rastogi et al. '449 system, the log mechanism of the present invention is local to the transaction control/processing sub-system for communications purposes and the log mirroring mechanism communicates with the log mechanism through a local, high speed datalink dedicated to that purpose. It is therefore in fact the system of the present invention, and not the system of Rastogi et al. '449, in which logged transaction information, that is, transaction state machines, can be recovered "immediately" from the log mechanism or log mirroring mechanism. It is also only in the system of the present invention, and not the system of Rastogi et al. '449, that can gain benefit of the "immediate" restoration of a transaction record from the log mechanism or log mirroring mechanism because it is only the system of the present that stores "state machines" that enable the restoration of a transaction at any point during the transaction.

It is therefore clear, and the belief and position of the Applicant, that the present invention is completely and fundamentally distinguished over and from the teachings of Rastogi et al. '449 under the requirements and provisions of 35 U.S.C. 102 and 35 U.S.C. 103 because Rastogi et al. '449 teaches and suggests only the capture and logging of transaction records representing the basic transaction operation commands, instructions and data and because Rastogi et al. '449 does not reach or even suggest in any way the capture, logging and restoration of transactions as sequences of state machines including the control and data values residing in and controlling the system at each step of the execution of a transaction.

It is further noted that the Examiner has in fact stated that Rastogi et al. '449 does not teach the capture, logging and restoration of transactions as sequences of state machines including the

control and data values residing in and controlling the system at each step of the execution of a transaction.

It is thus the belief and position of the Applicant that this distinction in itself is so fundamental and basic as to comprise a complete distinction of the present invention over and from the teachings and suggestions of Rastogi et al. '449 under the requirements of 35 U.S.C. 102 and 35 U.S.C. 103.

In further distinction between the present invention and the teachings of Rastogi et al. '449, however, it must be noted that the primary and secondary computer systems of the Rastogi et al. '449 system do not correspond either structurally or functionally with the dual control/processing sub-systems of the present invention. That is, and as described, for example, in Rastogi et al. '449 at column 3, line 8 through column 4, line 2, the two computer systems of the Rastogi et al. '449 system are not parallel, cooperating sub-systems within a system, but are completely and separate computer systems.

In still further fundamental distinction between the present invention and Rastogi et al. ' it must be noted that the dual control/processing sub-systems of the present invention operate independently but concurrently with each control/processing sub-system executing only the requests for transactions or operations that are directed to it, so that the full processing power of the two control/processing sub-systems is typically available at all times to process requests from clients. In basic contrast from the present invention, the two computer systems of the Rastogi et al. '449 system do not operate concurrently at any time. Instead, the primary processor alone operates to execute all transactions of the system while the secondary processor is always idle with respect to the execution of transactions, so that only one half the potential processing power of the Rastogi et al. '449 system is available at any time.

In addition, the primary and secondary computer systems of the Rastogi et al. '449 system do not correspond either structurally or functionally with the state machine logging mechanism and state machine log mirroring mechanism of the present invention as each of the primary and secondary computer systems of the Rastogi et al. '449 system is a full function, general purpose computer capable of performing both transaction operations and transaction logging. In contrast, the state machine logging mechanism and state machine log mirroring mechanism of the present invention are both dedicated purpose, specialized function mechanisms that are structurally and functionally different from one another and are directed to separate and distinctly different functions. In a like manner, the control/processing sub-system and a corresponding state machine log mirroring mechanism are structurally and functionally distinguished from the primary and secondary computer systems of the Rastogi et al. '449 system for the same reasons.

It must also be noted that a control/processing sub-system and its associated state machine logging mechanism with the associated state machine log mirroring mechanism cannot be compared, structurally or functionally, with the primary and secondary computer systems of the Rastogi et al. '449 system because the primary and secondary computer systems of the Rastogi et al. '449 system are in fact identical but completely separate and independent systems from one another. In contrast, the state machine log mirroring mechanism is functionally an integral element of the corresponding state machine logging mechanism, even through the state machine log mirroring system resides separately from the state machine logging mechanism as regards such support functions as the power supplies so as not be involved in a failure of the corresponding control/processing sub-system with which the state machine log generator and log reside.

It is the belief and position of the Applicant that for the reasons discussed above the teachings of Rastogi et al. '449 do not and cannot describe or suggest the present invention as

recited in claims 1, 3, 5, 9, 11, 13 and 15 under either or both of 35 U.S.C. § 102 or 35 U.S.C. § 103.

Lastly with regard to Rastogi et al. '449 in general, the Examiner had requested that the Applicant point out specifically the portions of Rastogi et al. '449's teachings that support the characterizations of Rastogi et al. '449 that have been described and discussed herein above. In reply, it will be noted that the Applicant has referred specifically to relevant portions of the teachings of Rastogi et al. '449 in the discussions herein above. For convenience, however, the Applicant can summarize the references to Rastogi et al. '449 as including column 8, lines 3-15; column 5, line 29 to column 7, line 15; column 8, lines 3-61; column 8, line 3 to column 11, line 8; column 3, line 22 to column 4, line 59; column 5, lines 9-44; column 7, line 38 to column 8, line 2; column 8, line 24 to column 10, line 30; and, column 3, line 8 to column 4, line 2.

Next considering certain specific issues raised by the Examiner with regard to Rastogi et al. '449 in the Final Office Action of January 18, 2005, in paragraph 8 of the Final Office Action of January 18, 2005 the Examiner disagrees with the Applicant's position and statements that Rastogi et al. '449 does not record or restore the intermediate sub-operations within a transaction. In support of this issue, the Examiner refers to column 8, lines 24-36 of Rastogi et al. '449 wherein Rastogi et al. '449 refers to a recovery algorithm that maintains a separate undo log and redo log in main memory for each transaction.

In response, the Applicant would like to first point out that the mere existence or non-existence of undo and redo logs says nothing in itself about what information is stored in the logs. In that respect, therefore, the portions of Rastogi et al. '449 cited by the Examiner in this paragraph actually have no information pertinent to the issue of whether or not Rastogi et al. '449 records or restores intermediate sub-operations within a transaction. The existence of undo and redo logs

merely means that Rastogi et al. '449 stores information to redo or undo a transaction at some level of detail, but not what that level of detail is or what the stored information is.

The significance of the undo and redo logs in the Rastogi et al. '449 must therefore be determined, if possible, from other portions of the Rastogi et al. '449 specification than just lines 24-36 of column 8. In this regard, however, and as will become apparent from the Rastogi et al. '449 specification and the following discussions, Rastogi et al. '449 essentially does not contain sufficient description of either the undo or redo logs or the information stored therein to determine the nature of the information stored therein, so that any conclusions about the redo and undo logs and the type of information logged in the Rastogi et al. '449 system must be deduced from the Rastogi et al. '449 specification as a whole.

For example, it must be noted that Rastogi et al. '449 does not even mention "state" or "state machine" in the meaning used in the present invention at any place in the specification, claims or drawings, so that Rastogi et al. '449 is clearly not storing system state or state machines in any form as Rastogi et al. '449 does not even refer to the concepts of operation state or state machines. More specifically, the only mention that Rastogi et al. '449 makes of the term "state" is when Rastogi et al. '449 is at column 3, lines 22-35, and at column 5, lines 9-65, wherein Rastogi et al. '449 defines the term "state" as bits of information stored in the primary and secondary systems that indicate, in each system, whether the system is the current primary system or is the current secondary system and whether the two computer systems are in "synchronization", that is, have matching copies of the primary computer system transaction log. That is, in the Rastogi et al. '449 system and at any given time only one of the two computer systems can be the current primary computer system, that is, the computer system that is actually executing system transactions while the other system is idle as a spare backup system with no function other than to store the committed "log" of completed transactions.

It will, therefore, be apparent to those of ordinary skill in the arts that in the teachings of Rastogi et al. '449 the term "state" has no relationship or meaning with regard to the state of operation of a system at each step in executing a transaction, or even to the actual execution of transactions, but instead relates only to the overall current functional assignments of the two systems and, in particular, to which one is assigned to execute transactions and which one is assigned to store a copy of the transaction log. "State" in Rastogi et al. '449 therefore has nothing to do with the actual execution of operations or transactions by the primary system, but only with respect to which system is executing the transactions.

Further in this regard, the fact that Rastogi et al. '449 employs the term "state" and explicitly describes the meaning of the word "state" in the specification of the Rastogi et al. '449 patent indicates very clearly that Rastogi et al. '449 did not even consider any other meanings of the term "state" other than that explicitly described in Rastogi et al. '449. In fact, the lack of any definition of the term "state" other than in the sense of identifying which system was primary and which system was secondary indicates quite clearly that Rastogi et al. '449 was not even aware in any way of the concept or meaning of "state" or "state machine" as employed in the present Application. This clearly further shows that Rastogi et al. '449 did not and could not intend to describe the logging of "state" or the use of "state machines" as employed in the present invention.

Considering other parts of the Rastogi et al. '449 specification for enlightenment concerning the significance of the undo and redo logs, column 7, line 49 through column 8, line 2, for example, describes that the Rastogi et al. '449 system does have a redo log and an undo log, and states that the system further includes a "dirty page table" that maintains a record of what memory pages have been updated since the last checkpoint, that is, the last time the memory and cache pages were brought up to date together. The description indicates, but does not describe in detail, that the operations of the "dirty page table" are associated in some way with the operations of the undo and

redo logs, that is, that the memory and cache management functions are somehow intertwined with the undo and redo logs.

At column 8, lines 3-15, Rastogi et al. '449 defines a "transaction" as being comprised of a sequence of operations at some level in the system and states that the Rastogi et al. '449 system records "transactions", but does not state whether the system records transactions at the "Lo" level, that is, at the level of the transaction itself, as an entity, or at some lower "Lz" level consisting of the sub-operations within a transaction. In this regard, it should be noted that a "transaction" may be undone or redone at any level, ranging from the level of the transaction itself to some level of sub-operation within the transaction, and that the information stored in the undo or redo log will dependent upon the level at which the transaction is redone or undone. At no point does Rastogi et al. '449 explicitly state the level of the transactions at which the redo and undo logs operate, so that this issue must be resolved, in so far as possible, from other aspects of the undo and redo log operations.

In this regard, Rastogi et al. '449 states in lines 15 through 68 of column 8 that transactions are handled and logged depending on whether the transactions are "precommit" or "commit" transactions. That is, in Rastogi et al. '449 a transaction is "precommit" when it has been stored in the system log in memory and is "commit" when it has been transferred from memory to the stable log, which is effectively on the hard disk of the primary system, where it may still be vulnerable. In this regard, the Rastogi et al. '449 the primary system transmits a copy of the primary system transaction records, that is, a log, to the secondary system through a network connection each time the transaction log in the primary system is "flushed" from the primary system memory space to the primary system mass storage device and when the primary system has transmitted a copy of the transaction log to the secondary system.

It is also described in lines 15 through 68 of column 8 that the operation of the undo and redo logs is intertwined with the cache/memory page update mechanisms in at least that updates to the cache and memory pages are recorded in the undo and redo logs. While the inter-operation of the cache/memory page update mechanisms and the undo/redo logs say nothing directly about the level of information the logs store, the discussion thereof by Rastogi et al. '449 provides some indication of the level of operation of the logs.

That is, and first considering the redo log and recovery mechanism, Rastogi et al. '449 states that a transaction "precommits", that is, is stored in the log in primary system memory when the first sub-operation of the transaction "pre-commits", that is, enters primary system memory to be executed. This statement indicates that the information that is committed to the log includes only the information available at the start of the operation and thereby cannot not include information occurring during the execution of the operation, such as sub-operation information, as that information has not even been brought into primary system memory, much less executed, at the time the transaction is committed to the log. This indicates that the information stored in the log does not include information generated during the execution of the transaction, that is, information from the execution of the sub-operations.

In this regard, it must be noted that the minimum information that is required to implement a redo mechanism is the initial command or instruction determining the transaction to be performed and the initial data used in the transaction, both of which are available when the transaction is initiated and before the first sub-operation is executed and which are sufficient allow a transaction to be redone from the start. The level of log information is therefore in agreement with the described operation of the pre-commit mechanisms, thereby indicating that the Rastogi et al. '449 system most probably does not store state or any other form of information derived from the execution of sub-operations within a transaction.

This conclusion is further supported when it is noted that while Rastogi et al. '449 does refer to the sub-operations of an operation or transaction when referring to the undo log, such as at column 8, lines 37-41, which indicates that the undo mechanism operates at the sub-operation level, at no point does Rastogi et al. '449 refer to the sub-operations of an operation or transaction when describing the redo log.

Next consider the undo log, it is possible that the undo log may store information regarding current operations of the system in greater detail than does the redo log. Rastogi et al. '449 states, for example, at column 10, lines 8-13, that when a transaction aborts the updates/operations described by the undo log are undone by traversing the undo log sequentially from the end and by executing, in reverse order, every undo record as if the execution were part of the transaction. Rastogi et al. '449 also refers to the sub-operations of an operation or transaction when referring to the undo log, such as at column 8, lines 37-41, which indicates that the undo mechanism operates at the sub-operation level.

It must be noted, however, that the undo log and recovery mechanism in the Rastogi et al. '449 system operate in a completely different manner than does the redo log and recovery mechanism and that the undo log is, in fact, not even a functional part of the operation logging/mirroring mechanism in the sense of a logging/mirror mechanism as described by the present invention or even in the sense described by Rastogi et al. '449. That is, Rastogi et al. '449 specifically states, for example, at column 8, lines 24-44, that the undo log for a transaction is deleted when the transaction "pre-commits". In other words, the undo log is not a part of a long term logging mechanism wherein the logs of transactions are stored for at least an extended period after the transaction is completed. The undo log and recovery mechanism is instead merely a short term, temporary buffer storage in memory that exists only during and before the execution of the transaction, which is typically adequate for the purposes of an undo log.

In summary, therefore, and for the reasons discussed above, it is the belief and position of the Applicant that Rastogi et al. '449 does not provide sufficient description of either the redo log or the undo log to comprise a teaching with regard to the present invention. It is further the belief and position of the Applicant that in so far as Rastogi et al. '449 contains a description of a redo log and mechanism or an undo log and mechanism that description supports the conclusion and position of the Applicant that the teachings of Rastogi et al. '449 pertinent to the Rastogi et al. '449 redo log are not pertinent to and do not disclose or even suggest the present invention under either 35 U.S.C. 102 or 35 U.S.C. 103.

In further summary, and again for the reasons discussed above, it is the belief and position of the Applicant that the teachings of Rastogi et al. '449 pertaining to an undo log are not pertinent to the present invention because the Rastogi et al. '449 undo log and recovery mechanism perform an entirely different function than does the log mechanism of the present invention.

It must also be noted that the present invention is further distinguished over and from the teachings of Rastogi et al. '449 because the present invention employs a single mechanism to support both the "redo" and "undo" functions. In fundamental contrast from the present invention, and as discussed in detail above, the Rastogi et al. '449 system requires two separate mechanisms to support these functions.

In paragraph 9 of the Final Office Action of January 18, 2005 the Examiner stated that although the Applicant used the term "sub-operation" in the arguments previously presented in discussing the distinctions of the present invention over Rastogi et al. '449, that express term does not appear in the claims themselves.

In response, the wishes to point out that the Applicant used the terms "sub-operation" in the discussions and arguments distinguishing the present invention over the cited prior art in an attempt to clarify the subject matter to the Examiner, believing that the term "sub-operation" might

be more familiar than the term "step", as in "step in the execution of an operation". As the Examiner will be aware from reviewing the specification of the present application, "step" is the term used in the specification and claims to refer to a sub-operation executed during and as part of the execution of an operation, such as a file transaction.

The Applicant apologizes for confusing the Examiner, and would request that the term "step" be read in place of the term "sub-operation" as it appears in the arguments discussing the distinctions of the present invention over the cited prior art, such as in "step in the execution of an operation".

It should be noted that the Applicant would prefer to retain the use of the term "step" to refer to a sub-operation in the execution of an operation as the term "step" is consistent with the terminology used to described the invention in the specification of the present invention. If the Examiner would prefer, however, the Applicant would be willing to amend the language of the claims to use the term "sub-operation" together with the explanatory statement "wherein a step is a sub-operation executed in the execution of an operation".

In conclusion, therefore, the limitations recited in the claims employ the term "step" to refer to a "sub-operation" executed in the execution of an operation and does so in order to conform to the terminology of the specification. It is therefore the belief and position of the Applicant that a limitation referring to a sub-operation in the execution of an operation does in fact appear in the claims.

In paragraph 10 of the Final Office Action of January 18, 2005, the Examiner objected to the Applicant's arguments distinguishing the present invention over the cited prior art and, in particular, with respect to the Applicant's statement that "in fundamental contrast from the typical from of "back-up" systems as taught by, for example, Rastogi et al. '449, the full processing power of the two control/processing sub-systems is available at all times to process requests from clients.

If one control/processing sub-system fails, the other control/processing sub-system continues operative to perform the request directed to it and to maintain the state machine information necessary to subsequently restore the failed control-processing sub-system." The Applicant stated therein that the system of the present invention was thereby distinguished over and from Rastogi et al. '449 because in the Rastogi et al. '449 system only one of the two systems comprising the total system was operative to process requests as the other of the two systems was essentially idle except for functioning as a log backup to the system processing requests.

The Examiner states that this argument is irrelevant since "this limitation doesn't appear anywhere in the claims".

In response, the Applicant would like to point out that this limitation expressly appears in claims 3 and 7 in the recitation of

"first and second control/processing sub-systems operating concurrently and in parallel, each including a file system processor performing file transaction operations in response to client requests directed to the first and second control/processing sub-systems and controlling file storage operations of the storage sub-system," and

in claims 11 and 15 in the recitation of

"the system resource including a system resource sub-system and first and second control/processing sub-systems, each including a system processor performing system resource operations in response to client requests directed to the first and second control/processing sub-systems and controlling operations of the system resource sub-system".

The Applicant therefore respectfully disagrees with the Examiner as this argument is therefore relevant to at least independent claims 3, 7, 11 and 15. Also, the Applicant believes that there is no requirement that a given argument distinguishing an invention over the prior art must

apply to every claim under consideration in order to be a relevant argument, and that claim 1 is fully distinguished over the cited prior art for other reasons that are fully discussed herein.

In paragraph 11 of the Final Office Action of January 18, 2005 the Examiner referred to the description by Rastogi et al. '449 at column 10, lines 52-61 regarding the roll-back of active transactions. The Examiner maintained that this means that all completed operations that have been directly invoked by the transaction or that have been directly invoked by an incomplete operation have to be rolled back, and that this means that the current state of a transaction is maintained.

In reply, the Applicant refers the Examiner to the above discussion of paragraphs 7 and 8 of the Final Office Action of January 18, 2005 and of the undo log and recovery mechanism.

Further in this regard, it should be noted that the operation of the undo mechanism as discussed herein above and as described by Rastogi et al. '449 does not conflict with the Examiner's conclusion that all completed operations that have been directly invoked by a transaction or that have been directly invoked by an incomplete operation have to be rolled back or that the current state of a transaction is maintained in order to do so.

As discussed herein above, however, the undo log and recovery mechanism in the Rastogi et al. '449 system operate in a completely different manner than does the redo log and recovery mechanism and that the undo log is, in fact, not even a functional part of the operation logging/mirroring mechanism in the sense of a logging/mirror mechanism as described by the present invention or even in the sense described by Rastogi et al. '449. That is, Rastogi et al. '449 specifically states, for example, at column 8, lines 24-44, that the undo log for a transaction is deleted when the transaction "pre-commits". In other words, the undo log is not a part of a long term logging mechanism wherein the logs of transactions are stored for at least an extended period after the transaction is completed. The undo log and recovery mechanism is instead merely a short

term, temporary buffer storage in memory that exists only during and before the execution of the transaction, which is typically adequate for the purposes of an undo log.

Considering the operation of the undo log and recovery mechanism in terms of operations that have been invoked during the execution of a completed or even uncompleted operation, it is apparent that such invoked operations are effectively part of the invoking operation and, as such, are logged on the undo log together with the invoking operation. Again, however, the undo log contains only the currently active operation, and any operations that are part of, that is, invoked by, the currently active operation. As such, any operations invoked by a completed but currently active operation or an uncompleted but currently active operation will be discarded with the active operation when the active operation is concluded in one way or another.

Again, therefore, and as stated herein above, it is the belief and position of the Applicant that the structure and operation of the undo log and recovery mechanism is irrelevant to the present invention as the undo log and mechanism are not part of a logging and mirror mechanism in the sense of either the present invention or even the redo log and mechanism of Rastogi et al. '449, but are an entirely different mechanism that may work in conjunction with a redo log and mechanism for certain specific purposes.

In paragraph 12 Final Office Action of January 18, 2005 the Examiner disagreed with the Applicant's previously submitted arguments that Rastogi et al. '449 does not have a logging mechanism capable of saving state at the sub-operation level, that is, at the step level, wherein a sub-operation, or a step, is an intermediate step executed in executing an operation.

In response, the Applicant refers the discussions herein above pertaining to, for example, paragraphs 7, 8 and 9 of the Final Office Action of January 18, 2005.

The Examiner also stated that the limitation of state logging for sub-operations does not appear in the claims.

In response, the Applicant refers the Examiner to the discussions herein above pertaining to paragraph 7, 8 and 9 of the Final Office Action of January 18, 2005.

The Examiner also stated that the limitations of the currently pending claims do not include the limitation, discussed in the previously submitted arguments distinguishing the present invention over Rastogi et al. '449, of continuous logging of state information at the at the sub-operation level, that is, at the step level. The Examiner maintained that the lack of this limitation allows the claims to be interpreted as stating that the logging operation occurs only once rather than for each sub-operation, or step, or an operation.

The Applicant believes that the Examiner is expressing what is essentially as 35 U.S.C. 112 issue as the Examiner seems of the opinion that the relevant language of the claims can be interpreted in more than one way.

The Applicant concurred that the claims do not include this limitation as an explicit recitation, and had amended the claims accordingly to remedy this lack. It will be noted that this amendment did not add any new matter to the specification or claims and does not alter the scope, meaning or subject matter of the claims, but has merely clarified the pertinent recitations.

Lastly in this regard, the Examiner had once again referred to the roll-back of active transactions as described in column 10, lines 52-61, although the Applicant was and is not sure of the point the Examiner is making by the reference at this point to the roll-back of active transactions. In response, however, the Applicant respectfully referred the Examiner to the discussions herein above of paragraphs 8, 9 and 11 of the Final Office Action of January 18, 2005.

In paragraph 13 of the Final Office Action of January 18, 2005 the Examiner disagreed with the Applicant's position that Rastogi et al. '449 does not teach any form of state machine representing control and data values in a state machine and again refers to column 8, lines 24-25 of Rastogi et al. '449.

In reply, the Applicant referred the Examiner to the Applicant's responses herein above to paragraphs 7, 8 and 11 of the Final Office Action of January 18, 2005.

In paragraph 14 of the Final Office Action of January 18, 2005 Final Action the Examiner has raised a number of issues, most of which are repetitive with the issues raised by the Examiner in preceding paragraphs of the Final Action. For example, the Examiner has again raised the issue that Rastogi et al. '449 discussed "state" wherein, in fact, the only mention that Rastogi et al. '449 makes of the term "state" is when Rastogi et al. '449 is at column 3, lines 22-35 and at column 5, lines 9-65. Rastogi et al. '449 therein defines the term "state" as bits of information stored in the primary and secondary systems that indicate, in each system, whether the system is the current primary system or is the current secondary system and whether the two computer systems are in "synchronization", that is, have matching copies of the primary computer system transaction log.

The Rastogi et al. '449 definition of the term "state" clearly has no relationship to the use of the terms "state" and "state machine" as defined in the present invention. This clearly indicates that Rastogi et al. '449 did not have concept of "state' or "state machine" as used in the present invention, and thereby could not and did not describe any type of system using "state" or "state machine" in the sense of the present invention.

Lastly, in paragraph 14 of the Final Office Action of January 18, 2005 the Examiner appears to quote a part of an argument previously advanced by the Applicant in support of the Examiner's position that the Applicant has acknowledged that the Rastogi et al. '449 system discloses the use of control and data values.

The quote from the Applicant's argument was taken out of context as the argument that is the source of the quote was advanced by the Applicant in support of the Applicant's position that although the Rastogi et al. '449 system does disclose control and data values, the Rastogi et al. '449 does not teach or suggest the use of state machines or the logging of state as in the present

invention. The "acknowledgment" cited by the Examiner, if the statement were to be taken in that sense, with which the Applicant does not agree, was that the Rastogi et al. '449 system does save and log control and data values as this is a necessary function in any logging system.

This is not in any way an acknowledgment or admission that the Rastogi et al. '449 system employs state and state machines in the manner of the present invention, but only that the Rastogi et al. '449 saves coarsely grained information regarding each operation performed by the Rastogi et al. '449 system. As discussed in the original argument, and as discussed herein above with regard to other paragraphs of the Final Action, it is the Applicant's position that the control and data values logged by the Rastogi et al. '449 are no more than the original instruction or command and input data initiating a given operation, not detail operation state information extracted and logged at each step of the execution of the operation.

In paragraph 15 of the Final Office Action of January 18, 2005 the Examiner once again disagreed with the Applicant's stated position that Rastogi et al. '449 does not teach or suggest anything beyond logging only the current instruction or command and input data, and that as a consequence Rastogi et al. '449 does not and cannot teach or suggest the logging of state or state machines, particularly at the sup-operation, or step, level. The Examiner again refers to column 8, lines 24-44 of Rastogi et al. '449 and the description of the Rastogi et al. '449 undo and redo log mechanisms.

In response, the Applicant refers the Examiner to the Applicant's above discussed responses to paragraphs 7, 8, 9 and 11 of the Final Office Action of January 18, 2005.

In paragraph 16 of the Final Office Action of January 18, 2005, the Examiner expressed disagreement with the Applicant's position that Rastogi et al. '449 does no and cannot capture or record the sub-operations within a transaction. The Examiner again referred to the discussions in Rastogi et al. '449 pertaining to the Rastogi et al. '449 redo and undo logs as teaching the capture

and recording of sub-operations. The Examiner further stated that the Examiner believes the Applicant's stated position to be merely an assumption made by the Applicant and asked the Applicant to point out specifically wherein in Rastogi et al. '449 the Applicant's position is supported.

In response, the Applicant referred the Examiner to the Applicant's responses to paragraphs 8, 9 and 11 Final Office Action of January 18, 2005 wherein the Applicant refers to specific sections of the Rastogi et al. '449 specification and explains in detail the Applicant's reasoning and conclusion to support the Applicant's position.

As also discussed herein above, it is the position of the Applicant that Rastogi et al. '449 does not in fact provide sufficient description of either the redo log or the undo log to comprise a teaching with regard to the present invention, or even of the Rastogi et al. '449 system, and is vague and inadequate even with regard to the Rastogi et al. '449 system itself.

In this regard, and for these reasons, the Applicant is of the impression that the Examiner is reading the teachings of the present invention onto and into the teachings of Rastogi et al. '449 to reach the conclusions expressed by the Examiner.

It is therefore the position of the Applicant that the Rastogi et al. '449 reference does not, in fact, comprise an adequate teaching under the requirements and provisions of either of 35 U.S.C. 102 or 35 U.S.C. 103.

It is further the belief and position of the Applicant, however, that in so far as Rastogi et al. '449 does contain a description of a redo and undo log and mechanism, the description supports the conclusion and position of the Applicant that the teachings of pertaining the Rastogi et al. '449 redo log and undo are not pertinent to and do not disclose or even suggest the present invention under either 35 U.S.C. 102 or 35 U.S.C. 103. For example, and again as discussed herein above, the teachings of Rastogi et al. '449 pertaining to an undo log are not pertinent to the present

invention because the Rastogi et al. '449 undo log and recovery mechanism perform an entirely different function than does the log mechanism of the present invention.

In paragraph 17 of the Final Office Action of January 18, 2005 the Examiner disagreed with a statement by the Applicant in the Response to the previous action to the effect that "It is noted, in this regard, that Rastogi et al. '449 does not in fact state that a log entry contains information taken during the execution of the transaction, but only that the stored information includes only the information essential to reconstruction of the transaction as an entity." The Examiner held that the information taken during execution of a transaction is the same as the information essential to reconstruction of the transaction and states that it is unclear how the information differs.

The Applicant believed that the Examiner has mis-read or mis-understood the statement is question and apologized for not making the meaning of the statement clear.

The essential meaning of this statement is that the type of information required to reconstruct a transaction, that is, to redo a transaction, depends upon the level at which the transaction is to be redone. If a transaction is to be redone, or re-executed, on a step-by-step basis, that is, by redoing each individual, specific sub-operation in the transaction, then it is usually necessary to record the pertinent information existing at the start of each step.

It is very common, however, to redo a transaction as an entity, that is, as a self-contained operation wherein the transaction is treated as a self-contained entity rather than as a sequence of individuals steps, or sub-operations, and it should noted that the statement in question refers to redoing "the transaction as an entity". This is the method typically used in older systems of the prior art, such as Rastogi et al. '449, because it requires the recordation of much less information than does a step-by-step method. That is, the redoing of a transaction as an entity typically requires the recordation of only the initial command or instruction stating the transaction to be performed and the initial input data to the transaction. This information is all that was required for

a system to "do" the transaction the first time, and is all that a system typically requires to redo, or re-execute, a transaction as the system will thereafter follow its programming to execute, or redo, the transaction, just as in the initial execution of the transaction.

It should also be noted, for the sake of completeness, that as discussed herein above with regard to paragraphs 8, 9 and 11 of the Final Office Action of January 18, 2005, the "redo" of a transaction is a different type of operation than the "undo" of a transaction and may therefore require a different level of information than does the "undo" of a transaction. As also discussed herein above, however, the teachings of Rastogi et al. '449 pertaining to an undo log are not pertinent to the present invention because the Rastogi et al. '449 undo log and recovery mechanism perform an entirely different function than does the log mechanism of the present invention. For example, the Rastogi et al. '449 undo log and mechanism are a short term log that stores information pertaining to a transaction only during the actual execution of the transaction and discards the information as soon as the transaction is completed, which is in complete contrast to both the present invention and even the redo log and mechanism of the Rastogi et al. '449 system itself.

In paragraph 18 of the Final Office Action of January 18, 2005, the Examiner disagreed with the Applicant's statement that Rastogi et al. '449 does not even mention restoring or resuming a transaction at any of the sub-operations, or steps, comprising the transaction but only redoing or undoing as transaction, and again refers to column 10, lines 52-61 of Rastogi et al. '449.

In response, the Applicant referred the Examiner to the above responses to paragraphs 8, 9 and 11 of the Final Office Action of January 18, 2005, wherein the Applicant discusses and explains in detail the difference between the Rastogi et al. '449 undo and redo logs and mechanisms and the transaction information recorded therein and the logs of the present invention

wherein information is recorded for each sub-operation, that is, state and state machine, of a transaction.

In addition, the Examiner also stated that this argument is irrelevant because the limitation does not appear anywhere in the claims.

The Applicant cannot ascertain from the Examiner's statement whether the Examiner is saying that the limitations of restoring or resuming a transaction is not recited in the claims or that the limitation of recording information for each sub-operation or step of a transaction is not recited in the claims. The Examiner is incorrect in either case, however.

First, the Applicant would like to point out that each of the claims contains recitations directed to the extraction, or capture, and recording, or storing, of information for each sub-operation or step of a transaction wherein that information is represented by the system state for the sub-operation or step, which is in turn represented as a state machine for the sub-operation or step.

Secondly, the Applicant must also point out that each of the claims contains recitations reciting the restoration of transactions from the state machine information stored in the log and mirror logs and each of the claims contains recitations, either directly or by dependence, that the state machine information, that is, the state information contained in each state machine, pertains to a corresponding sub-operation or step of a transaction. As a consequence, each claim contains recitations of the restoration or resumption of a sub-operation or step of a transaction.

For these reasons, therefore, it is the belief and position of the Applicant that these arguments are pertinent to the present invention as recited in the claims and distinguished the present invention as recited in the claims over the teachings and suggestions contained in Rastogi et al. '449.

In paragraph 19 of the Final Office Action of January 18, 2005 the Examiner questioned the Applicant's argument regarding the two computers comprising the Rastogi et al. '449 system and rejects the Applicant's arguments regarding this issue.

In this regard, it should be noted that the Examiner cited U.S. Patent No. 5,781,716 to Hemphill et al.in rejection of the Applicant's argument, but not in any rejection of the claims under either of 35 U.S.C. 102 or 35 U.S.C. 103.

In brief, the Applicant's argument is that in Rastogi et al. '449 the two computers are not parallel systems wherein both systems are concurrently engaged in the processing of transactions, that is, wherein each processes its own transactions. In the Rastogi et al. '449 system the two computers are separate and, at any given time, perform completely different functions. That is, only one system is actually processing transactions at any given time while the other system serves only to store the transaction log for the active system.

In a system of the present invention as recited, for example, in claims 3, 7, 11 and 15, the system is comprised of two sub-systems that normally operate concurrently and parallel, that is, with each executing its own transactions, which is a fundamentally different system from Rastogi et al. '449 wherein only one computer actually processes transactions at a time.

The Applicant has also reviewed Hemphill et al. '716, however, which was newly cited by the Examiner, but not in rejection of the claims, and concurs that Hemphill et al. '716 does teach a system comprised of two sub-systems that normally operate concurrently and parallel, that is, with each executing its own transactions.

Therefore, and while Rastogi et al. '449 is not prior art with respect to this aspect of a system of the present invention, and Hemphill et al. '716 is pertinent this aspect of a system of the present invention, but does not appear to teach any other aspects of the present

invention so that the present invention as recited under claims 1, 3, 5, 7, 9, 11, 13 and 15 remains patentably distinguished over the prior art for others of the reasons discussed above.

In conclusion, therefore, it is the Applicant's belief that the rejection of claims 1, 3, 5, 9, 11, 13 and 15 as unpatentable as anticipated under 35 U.S.C. § 102 over Rastogi et al. '449 is in error and that claim 1 and thus claims 3, 5, 7, 9, 11, 13 and 15 are patentably distinguished over the cited prior art under the requirements and provisions of 35 U.S.C. 102.

In particular, Rastogi et al. '449 does not teach or suggest under the requirements and provisions of 35 U.S.C. 102 the recitations of claim 1, and thus of claims 3, 5, 7, 9, 11, 13 or 15, of:

> "a state machine logging mechanism operating concurrently and cooperative with the file system processor, including

a state machine log generator for extracting state machine information defining at least one state machine during an execution of an operation, the at least one state machine representing a current state of execution of a file transaction wherein a state machine is comprised of state information including control and data values representing a state of operation of the control/processing sub-system at a given time, wherein

the control/processing sub-system is a state machine system defined during the execution of an operation by at least one sequential state machine defined by a state of operation of the state machine system during a step in the execution of the operation, and wherein

a file transaction operation is represented by at least one_sequential state machine wherein each state machine is defined by data and control values residing in the state machine system during existence of state machine of the sequence, and

a state machine log for storing the state machine information, wherein the state machine log generator is responsive to the restoration of operation of the file server after a

failure of file server operations for reading the state machine information from the state machine log and restoring the state of execution of a file transaction."

(d) It is the Applicant's belief that the rejection of claims 2, 4, 6, 8, 10, 12, 14 and 16 under 35 U.S.C. § 103(a) over Rastogi et al. '449 and further in view of U.S. Patent No. 5,513,314 to Kandasamy et al. for a FAULT TOLERANT NFS SERVER SYSTEM AND MIRRORING PROTOCOL, hereafter referred to as "Kandasamy et al. '314", is in error for the following reasons.

First, it must be noted that claims 2, 4, 6, 8, 10, 12, 14 and 16 are each dependent from a corresponding one of claims 1, 3, 5, 7, 9, 11, 13 and 15 and that claims 3, 5, 7, 9, 11, 13 and 15 each include recitations and limitations identical or very similar to those of claim 1. Claims 2, 4, 6, 8, 10, 12, 14 and 16 thereby incorporate the recitations and limitations of claim 1 or, more specifically, the corresponding one of claims 1, 3, 5, 7, 9, 11, 13 and 15. Claims 2, 4, 6, 8, 10, 12, 14 and 16 are thereby distinguished over and from the cited prior art, specifically Rastogi et al. '449, for the same reasons that claim 1 is distinguished over and from the cited prior art.

In addition, claims 2, 4, 6, 8, 10, 12, 14 and 16 each add recitations and limitations directed to a log mirroring system to the recitations and limitations of the corresponding ones of claims 1, 3, 5, 7, 9, 11, 13 and 15, thereby providing further grounds by which claims 2, 4, 6, 8, 12, 14 and 16 are distinguished over the cited prior art.

In as much as Rastogi et al. '449 has been discussed in detail under the arguments pertaining to Issue (c), Rastogi et al. '449 will not be discussed again in full detail with respect to claims 2, 4, 6, 8, 10, 112, 14 and 16 as, for the reasons discussed just above, claims 2, 4, 6, 8, 10, 12, 14 and 16 are fully distinguished over and from Rastogi et al. '449 for the reasons discussed herein above with regard to Issue (c). The Applicant accordingly respectfully requests that the

above discussions regarding Issue (c) be regarded as incorporated into the following discussions by reference rather than repeated.

The following will therefore first discuss the teachings of Kandasamy et al. '314 and the relevance of Kandasamy et al. '314 to the present invention as recited in claims 2, 4, 6, 8, 10, 12, 14 and 16, and certain specific issues raised by the Examiner in the Office Action of January 18, 2005, the Applicant's responses to those issues having been presented in the Response of March 7, 2005. The following will then discuss the relevance of Rastogi et al. '449 in view of Kandasamy et al. '314 to the present invention.

Kandasamy et al. '314 describes a fault tolerant mirroring file server system in which one or more clients and two or more file servers are interconnected through a local area network and network protocol layers. The file servers include a control protocol with asymmetric responses such that a request for a file transfer from a client to a first file server is also received by and executed, or replicated, by the second file server and a request for a file transfer from a first file server to the client is normally executed by the first file server but is responded to by the second file server if the first file server does not respond.

It is apparent that the Kandasamy et al. '314 system will provide faster backup, mirroring and restoration than will the Rastogi et al. '449 simply because the Kandasamy et al. '314 system performs data transfers from the client to both servers in parallel while the Rastogi et al. '449 operates sequentially by writing to the primary server and having the primary server subsequently write a backup copy to the secondary server. It should also be noted that reads of data from one server to the client will occur at about the same speed in both systems because in both systems the backup server will wait to determine whether the primary server has responded or has failed before responding to a read request with the backup copy.

A review of the teachings of Kandasamy et al. '314, such as at column 3, lines 8-31, column 5, line 51 through column 6, line 19, and column 10, line 50 through column 11, line 10, shows that Kandasamy et al. '314 is similar to Rastogi et al. '449, and is thus distinguished from the present invention for the same reasons, in that the Kandasamy et al. '314 system captures, records and backs up data transfer requests only as self-contained entities. Stated another way, the Kandasamy et al. '314 system is in basic contrast from the present invention because the Kandasamy et al. '314 system does not and cannot capture and record the sub-operations comprising the data transfer requests, but instead captures and records data transfer requests only as entities in themselves.

In this regard, it will be noted that the teachings of Kandasamy et al. '314 referred to above by column and line consistently refer to data transfer requests as self-contained entities and consistently describe each data transfer request as being comprised of a request and the associated data, which comprises a complete operation in itself. There is no suggestion in Kandasamy et al. '314 that an operation or transaction may be comprised of a sequence of data transfer requests or that a data transfer request may be comprised of a sequence of sub-operations.

Further in this regard, it must be noted that Kandasamy et al. '314 describes the clients and the first and second servers as being interconnected by and communicating through a local area network using a common and widely used multi-layer communications protocol. The communications protocol in turn imposes constraints on the form the communications can take and the rate at which the communications can occur. More specifically, the described communications protocol demands the use of communications "packets" wherein each packet is self-contained and complete entity in itself, including an address, and command or instruction that may pertain to or be the contents, and possibly a certain amount of data.

While this packet structure does not demand that each data transfer request be contained in a single packet, it does accommodate communications in which each data transfer request is a self-contained entity much more efficiently and readily than it does other forms of communication, such as streams of state engines wherein each state engine resides in a packet and the length of the stream is essentially unknown.

It must be further noted that the necessity to pass each communication through the multi-layer protocol at each end of the communication path, such as between a client and a data server, also imposes certain speed constraints on the Kandasamy et al. '314 system; that is, the system can transfer only so many "packets" per unit time regardless of the contents of the packets. This limitation therefore strongly biases the Kandasamy et al. '314 system to the method of operation in which each data transfer request is contained in as few packets as possible. This in turn strongly influences the Kandasamy et al. '314 system to the method of operation wherein each data transfer request is a self-contained and complete operation rather than a sequence of sub-operations of unknown length.

For this reason, therefore, the communications method employed in the Kandasamy et al. '314 system reflects and strongly supports the conclusion that each data transfer request is a self-contained operation rather than a sequence of sub-operations, or states, or unknown length.

As discussed above, this conclusion is strongly supported by the fact that Kandasamy et al. '314 consistently refers to data transfer requests as self-contained entities and consistently describes each data transfer request as being comprised of a request and the associated data, which comprises a complete operation in itself. There is no suggestion in Kandasamy et al. '314 that an operation or transaction may be comprised of a sequence of data transfer requests or that a data transfer request may be comprised of a sequence of sub-operations.

It is the belief and position of the Applicant that for the reasons discussed above the teachings of Kandasamy et al. '314 do not and cannot describe or suggest the present invention as recited in the claims under either or both of 35 U.S.C. § 102 or 35 U.S.C. § 103.

Continuing with Kandasamy et al. '314, and as discussed herein above, the Examiner suggested that Kandasamy et al. '314 teaches a system similar to that of the present invention in that the first and second data servers comprise separate and independent but concurrently operating units that provide "instantaneous" responses to the clients or to a failure in one of the units. The Applicant disagrees for a number of reasons.

First, and as described above and in Kandasamy et al. '314, Kandasamy et al. '314 describes a fault tolerant mirroring file server system with asymmetric responses such that a request for a file transfer from a client to a first file server is concurrently received and executed by the second file server and a request for a file transfer from a first file server to the client is normally executed sequentially by the two servers, first by the first file server in direct response to the request and subsequently by the second file server if the first file server does not respond. While the Kandasamy et al. '314 does have a concurrent mode of operation between the two servers for write requests, the Kandasamy et al. '314 system is equally a sequential system as regards responses to read requests.

The system of the present invention is thereby fundamentally distinguished from the teachings of Kandasamy et al. '314 in that the control/processing sub-systems of the present invention operate concurrently and in parallel at all times rather than for only certain specific operations.

Further in this regard, it must be noted that the dual control/processing sub-systems of the present invention operate concurrently with one another but independently of one another with each control/processing sub-system executing only the requests for transactions or operations that

are directed to it, so that the full processing power of the two control/processing sub-systems is typically available at all times to process requests from clients. In contrast from the system of the present invention, the two data servers in the Kandasamy et al. '314 system both always respond to and operate upon the same data transfer requests. As a consequence, and while the two file servers of the Kandasamy et al. '314 system operate concurrently, they do not and cannot operate independently from one another. In fact, any attempt to cause the two data file servers of the Kandasamy et al. '314 system to operated independently from one another, such as on different data transfer requests at the same time, would be in direct contradiction to the fundamental principles of operation taught by Kandasamy et al. '314. Stated another way, the Kandasamy et al. '314 is operative as described and taught only when the two data file servers operate concurrently but in a first/second server relationship wherein both servers operate on the same data transfer request at the same time.

The system of the present invention is thereby fully and completely distinguished over and from the teachings of Kandasamy et al. '314 in that the control/processing sub-systems of the present invention operate concurrently and in parallel but independently from one another, with each operating separately on the data requests addressed to it, while the data servers of the Kandasamy et al. '314 system must operate together on each and every data transfer request.

Lastly, it must be noted that the first and second data servers of the Kandasamy et al. '314 system do not correspond either structurally or functionally with the state machine logging mechanism and state machine log mirroring mechanism of the present invention as the first and second data servers of the Kandasamy et al. '314 system are identical but each is a separate, self-contained general purpose data file system. In contrast, the state machine logging mechanism and state machine log mirroring mechanism of the present invention are both dedicated purpose, specialized function mechanisms that are structurally and functionally different from one another

and are directed to separate and distinctly different functions. In a like manner, and in complete contrast from the two identical and separate data file servers of the Kandasamy et al. '314 system, the state machine log mirroring mechanism of the present invention is functionally an integral element of the corresponding state machine logging mechanism, even through the state machine log mirroring system resides physically separate from the state machine logging mechanism as regards such support functions as the power supplies.

The system of the present invention is thereby still further fundamentally distinguished from the teachings of Kandasamy et al. '314 for the reasons just discussed.

In conclusion, therefore, it is the belief and position of the Applicant that for the reasons discussed above the teachings of Kandasamy et al. '314 do not and cannot describe or suggest the present invention as recited in the claims under either or both of 35 U.S.C. § 102 or 35 U.S.C. § 103.

Next considering certain detailed issue raised by the Examiner in the Final Office Action of January 18, 2005, in paragraphs 20 and 21 thereof the Examiner raised essentially the same issues that the Examiner raised in paragraph 19 of the Final Office Action of January 18, 2005, and rejects the Applicant's previous arguments on this issue.

In response, the Applicant refer to the Applicant's above discussed reply to paragraph 19, of the Final Office Action of January 18, 2005, which addressed these same issues with respect to Rastogi et al. '449.

In paragraphs 22, 23, 24 and 25 of the Final Office Action of January 18, 2005 the Examiner essentially addresses the same issues, but in progressively greater detail. In particular, the Examiner addresses whether Kandasamy et al '314 teaches the capture and recording of data transfer requests as sub-operations or only as entities, disagrees with the Applicant's arguments, states that the Applicant's arguments are merely assumptions made by the Applicant, and states

that the Examiner cannot find any support for the Applicant's arguments in the portions of Kandasamy et al '314 previously cited by the Applicant.

To consider the Examiner remarks in paragraphs 22, 23, 24 and 25 in greater detail, the Applicant referred the Examiner to column 3, lines 5-25, column 5, line 30 to column 6, line 20, column 7, line 22 through column 8, line 31,column 9, line 19 to column 11, line 56 of Kandasamy et al. '314. The following remarks are taken almost verbatim from Kandasamy et al. '314, and the conclusions drawn from the descriptions provided by Kandasamy et al. '314 are not mere "assumptions" drawn by the Applicant but are obvious conclusions that will be well understood and accepted by those of ordinary skill in the arts

To begin with, the Applicant has explained that Kandasamy et al. '314 describes a fault tolerant mirroring file server system in which one or more clients and two or more file servers are interconnected through a local area network and network protocol layers using packet communications. In particular, a request for a file transfer from a client to a file server is received by and executed, or replicated, by both file servers, but the initially addressed file server will normally respond to the request while the other file server will respond if the initially addressed server does not. The file servers thereby operate in parallel and concurrently to handle each read or write request, although only one will actually transfer the read or write data to or from the client while the other replicates the data transfer, so that the data stored in the two servers is identical and so that either can therefore reply to any read or write request.

There are therefore a number of aspects of the Kandasamy et al. '314 system that must be considered.

First, it must also be noted that Kandasamy et al. '314 does not describe the system as extracting, capturing or storing any form of state or state machine information representing a data transfer, and in fact does not even mention state or state machines, but instead describes the

system only as storing a copy of the data transfer itself, that is, a copy of the transfer request and the data actually transferred. Unlike the present invention, therefore, the Kandasamy et al. '314 does not capture, extract or record state or state machine information.

In further distinction between the present invention and Kandasamy et al. '314, it must be noted that in the Kandasamy et al. '314 system the records stored by the two file servers are actual and literal copies of the data transfers themselves. That is, each record includes the read or write request, a copy of the data that is actually transferred, and, usually, some acknowledgment that the transfer was completed. Again, therefore, and in complete support of the above "conclusion", the Kandasamy et al. '314 does not extract, capture or record state information or state machines in any form, but merely stores direct copies of each transfer request and the corresponding data that is transferred.

It addition, it must be noted that each data transfer is therefore essentially treated as a complete and self-contained entity comprised of at least the request and the data transferred. This conclusion is not only supported by the direct description provided by Kandasamy et al. '314, but is also supported in that it is apparent that the elements of the Kandasamy et al. '314 system involved and active in a data transfer operation include not only the requesting client but both file servers, all of which are completely and concurrently involved in a transfer at the same time, so that each transfer must be dealt with as an entity.

A further aspect of the Kandasamy et al. '314 system that must be noted is that Kandasamy et al. '314 describes the communications between the clients and the two file servers as being through a packet type communications network. That is, and as is well known in the arts, a packet type communication system divides the information to be transferred between a client and a file server, that is, the transfer request, the data to be transferred, and any acknowledgment/coordination/synchronization information, into packets of fixed size and format.

The packet communication system then sequentially transfers as many packets as are required to contain the transfer request, the data to be transferred, and any acknowledgment/coordination/synchronization information.

While it is possible and common to interleave packet transfers between different parties on a communication network, it is normally necessary to execute a packet transfer between two parties, that is to sequentially transfer the number of packets necessary to contain the information being transferred between the parties, as a single operation to avoid confusion among the packets arriving at the receiving party. This again further supports the conclusion that each data transfer in the Kandasamy et al. '314 system is completed as a single entity before a next data transfer is initiated.

It is therefore again the belief and position of the Applicant that Kandasamy et al. '314 has no teachings or suggestions that are relevant to the present invention under either of 35 U.S.C. 102 or 35 U.S.C. 103.

Considering certain of the Examiner's statements from paragraphs 22, 23, 24 and 25 of the Final Office Action of January 18, 2005 in further detail, in paragraph 22 the Examiner states that Kandasamy et al. '314 describes that the transactions between the systems are concurrent, so that sub-operations would be captured.

It has been defined in the present Application that sub-operations, or steps, are the lower level operations that comprised a higher level operation and that are carried out in executing the higher level operation. Concurrent operation, however, refers to the case when to systems or sub-systems or elements of a system operate at the same time to perform the same or similar operations.

There is therefore no connection or relationship between performing operations "concurrently" and performing "sub-operations" as these are two entirely different matters. The

Applicant therefore respectfully disagrees with the Examiner's conclusion that "concurrently" implies the recording of "sub-operations" and feels that this conclusion is unsupported by Kandasamy et al. '314. The Applicant cannot address the issue further, however, as the Examiner has not explained the reason behind this conclusion but has simply stated the conclusion.

In paragraph 23 of the Final Office Action of January 18, 2005 the Examiner disagrees with the Applicant's arguments regarding the operation of the Kandasamy et al. '314 as a packet communication system and the Applicant's conclusion that this operation means that data requests and transfers are executed a self-contained operations, each including a transfer request and a data transfer.

In reply, Kandasamy et al. '314 clearly describes the system as employing packet communications and as executing data transfers between file servers and clients. The conclusions that the Applicant draws from these descriptions in Kandasamy et al. '314 are not unsupported, but are commonly known aspects of both data transfers and packet communications systems, which the Applicant has again attempted to explain above. Again, however, the Applicant cannot address this issue further as the Examiner has merely stated a disagreement with the Applicant's conclusions and has not expressed any reasoning to support the disagreement with the Applicant.

In paragraph 24 of the Final Office Action of January 18, 2005 the Examiner has stated that the Examiner does not understand the relevance behind the Applicant's statement that the invention is distinguished from Kandasamy et al. '314 because the system of the present invention operates concurrently and in parallel at all times.

As discussed herein above, a system of the present invention as recited, for example, in claims 3, 7, 11 and 15, is comprised of two sub-systems that normally operate concurrently and parallel, that is, with each executing its own transactions at the same time the other is executing its own transactions.

In the Kandasamy et al. '314 system, however, and as also discussed above, each file server replicates, that is, copies, each data transfer operation between the other filer server and a client. In a given data transfer operation, therefore, the requesting client and both file servers are fully involved in the transfer operation at the same time, with one file server performing the data transfer and the other replicating the data transfer. As a consequence, the two file servers cannot operate on different data transfer operations at the same time and thereby cannot execute two different operations concurrently and in parallel, as can the system of the present invention.

It is therefore apparent that this distinction is substantive and significant.

In paragraph 25 of the Final Office Action of January 18, 2005 the Examiner refers to the concurrent operation of two sub-processors in the present invention as discussed just above with reference to paragraph 24 and a previously discussed with reference to paragraph 19.

The Examiner stated that the Examiner would not address this issue because the limitation of two concurrently operating, parallel sub-processors does not appear anywhere in the claims.

In response, the present and operation of parallel, concurrently operating control/processing sub-systems is explicitly recited in, for example, claims 3, 7, 11 and 15, and thereby in their respective dependent claims, so that this argument is pertinent and material and, among other distinctions, distinguishes the present invention as claimed over the cited references.

Next considering the combination of Rastogi et al. '449 in further view of Kandasamy et al. '314, it is the belief and position of the Applicant that the combination of Rastogi et al. '449 with Kandasamy et al. '314 would not be apparent to those of ordinary skill in the relevant arts because of the very different fundamental nature and operation of the two systems and the undesirable consequences of forming such a combination. That is, the Rastogi et al. '449 system is a sequentially operating system wherein a data read or write transaction is first sent from the client to the primary server and is then subsequently backed up by transfer of a backup copy of the

transaction from the primary server to the secondary server when the transaction has completed in the primary server. In fundamental contrast from Rastogi et al. '449, the Kandasamy et al. '314 is a parallel, concurrent system for data transfers from a client to the servers; that is, the data write request and the data is transferred from the client and to both servers at the same time and both servers execute the data write request in parallel.

As such, the combination of Rastogi et al. '449 with Kandasamy et al. '314 would require the combination of two systems having completely and fundamentally different and conflicting modes of operation and would not result in any form of feasible system. As a consequence, the modification of Rastogi et al. '449's sequential system by the addition of the parallel operation taught by Kandasamy et al. '314 would require more than a "modification" to the Rastogi et al. '449 system and would require what would amount to a complete reconstruction of the Rastogi et al. '449 as a Kandasamy et al. '314 system. That is, and for example, the addition of the Kandasamy et al. '314 parallel method of operation to the Rastogi et al. '449 sequential system would result in nothing more than the Kandasamy et al. '314 system rather than in some form of hybrid sequential/parallel system.

Assuming solely for the purposes of discussion, and without any admission or agreement as regards the validity of such a combination, that some combination of Rastogi et al. '449 with Kandasamy et al. '314 were to be achieved, the result would be no more than a form of the Kandasamy et al. '314 system as discussed herein above. As such, the present invention as recited in the claims would be fully and patentably distinguished over and from the teachings of some combination of Rastogi et al. '449 with Kandasamy et al. '314 for the same reasons that the present invention is fully and patentably distinguished over and from the teachings of Kandasamy et al. '314.

It is therefore the belief and position of the Applicant that the combination of Rastogi et al. '449 in further view of Kandasamy et al. '314 is not a valid combination or teaching and, for the reasons discussed above, would not in any instance teach or even suggest the present invention as recited in the claims to those of ordinary skill in the arts under the requirements and provisions of 35 U.S.C. 103.

Finally considering certain detailed issues raised by the Examiner in the Final Office Action of January 18, 2005 regarding the combination of Rastogi et al. '449 in further view of Kandasamy et al. '314, in paragraph 26 of the Final Office Action of January 18, 2005 the Examiner expressed disagreement with the Applicant's position that the combination of Rastogi et al. '449 and Kandasamy et al. '314 is invalid because, as stated by the Examiner, the combination of Kandasamy et al. '314 with Rastogi et al. '449 would be an obvious improvement over Rastogi et al. '449 alone. The Examiner, however, does not show or suggest how the Rastogi et al. '449 and Kandasamy et al. '314 references could be combined to provide a resulting system having the claimed advantages.

The Applicant therefore continues to respectfully disagree with the Examiner for the reasons stated previously. Briefly, the systems taught by Rastogi et al. '449 and Kandasamy et al. '314 are too different in structure and principle of operation to be a valid combination and, as also previously discussed in detail, any possible combination of Rastogi et al. '449 and Kandasamy et al. '314 would result in a system not only not having the perceived advantage, but one have numerous additional disadvantages.

Stated another way, when combining references it is not sufficient only to perceive some advantage from the combination, it is also necessary to show some teaching in the references pertinent to how to combine the references into a working system having the claimed advantages. The Examiner has failed to make this showing, while the Applicant has described in detail the

results that may accrue from such a combination and the very real disadvantages resulting from such a combination.

In summary, therefore, and for the reasons discussed herein above with respect to the Examiner's remarks, it is the belief and position of the Applicant that the present invention as recited in the claims as amended herein above is fully and patentably distinguished over and from the teachings of Rastogi et al. '449, Kandasamy et al. '314 and all combination thereof under the requirements and provisions of both 35 U.S.C. 102 and 35 U.S.C. 103.

In conclusion, therefore, it is the Applicant's belief that the rejection of claims 2, 4, 6, 8, 10, 12, 14 and 16 as unpatentable as anticipated under 35 U.S.C. § 102 over Rastogi et al. '449 in further view of Kandasamy et al. '314 is in error and that claim 2 and thus claims 4, 6, 8, 10, 12, 14 and 16 are patentably distinguished over the cited prior art under the requirements and provisions of 35 U.S.C. 103.

In particular, Rastogi et al. '449, Kandasamy et al. '314 and Rastogi et al. '449 in further view of Kandasamy et al. '314 do not teach or suggest under the requirements and provisions of 35 U.S.C. 103 the recitations of claim 1 that are incorporate into claim 2 and the recitations of claim 2, and thus of claims 4,6, 8, 10, 12, 14 and 16 of, from claim 1:

> "a state machine logging mechanism operating concurrently and cooperative with the file system processor, including
>
> a state machine log generator for extracting state machine information defining at least one state machine during an execution of an operation, the at least one state machine representing a current state of execution of a file transaction wherein a state machine is comprised of state information including control and data values representing a state of operation of the control/processing sub-system at a given time, wherein

the control/processing sub-system is a state machine system defined during the execution of an operation by at least one sequential state machine defined by a state of operation of the state machine system during a step in the execution of the operation, and wherein

a file transaction operation is represented by at least one_sequential state machine wherein each state machine is defined by data and control values residing in the state machine system during existence of state machine of the sequence, and

a state machine log for storing the state machine information, wherein the state machine log generator is responsive to the restoration of operation of the file server after a failure of file server operations for reading the state machine information from the state machine log and restoring the state of execution of a file transaction"

or, from claim 2:

"a state machine log mirroring mechanism that is functionally integral with the state machine log and that operates concurrently and in parallel with the state machine log and that is located separately from the control/processing sub-system and communicating with the state machine log generator for receiving and storing mirror copies of the state machine information, wherein the state machine log mirroring mechanism is responsive to the restoration of operation of the file server after a failure of file server operations for reading the state machine information from the state machine log mirroring mechanism and restoring the state of execution of a file transaction".

Lastly, the Applicant offers, under the same arguments of allowability as present above, the entry of the amended claims of Appendix B should it be held that the terminology in the present claims pertaining to sequences of state machines is unclear.

In the event that there are any fee deficiencies or additional fees are payable, please charge the same or credit any overpayment to our Deposit Account (Account No. 04-0213).

Respectfully submitted,

Gary D. Clapp, Reg. No. 29,055
Customer No. 020210
Davis & Bujold, P.L.L.C.
Fourth Floor
500 North Commercial Street
Manchester NH 03101-1151
Telephone 603-624-9220
Facsimile 603-624-9229
E-mail: patent@davisandbujold.com

APPENDIX A

PENDING CLAIMS

1. A file server performing file transaction operations in response to file transaction requests by the clients and including a state machine logging mechanism, comprising:

a storage sub-system, and

a control/processing sub-system including

a file system processor performing file transaction operations in response to client requests and controlling file storage operations of the storage sub-system, and

a state machine logging mechanism operating concurrently and cooperative with the file system processor, including

a state machine log generator for extracting state machine information defining at least one state machine during an execution of an operation, the at least one state machine representing a current state of execution of a file transaction wherein a state machine is comprised of state information including control and data values representing a state of operation of the control/processing sub-system at a given time, wherein

the control/processing sub-system is a state machine system defined during the execution of an operation by at least one sequential state machine defined by a state of operation of the state machine system during a step in the execution of the operation, and wherein

a file transaction operation is represented by at least one_sequential state machine_wherein each state machine is defined by data and control values residing in the state machine system during existence of state machine of the sequence, and

a state machine log for storing the state machine information, wherein the state machine log generator is responsive to the restoration of operation of the file server after a failure of file server operations for reading the state machine information from the state machine log and restoring the state of execution of a file transaction.

2. The file server of claim 1, wherein the state machine logging mechanism further comprises:

a state machine log mirroring mechanism that is functionally integral with the state machine log and that operates concurrently and in parallel with the state machine log and that is located separately from the control/processing sub-system and communicating with the state machine log generator for receiving and storing mirror copies of the state machine information, wherein the state machine log mirroring mechanism is responsive to the restoration of operation of the file server after a failure of file server operations for reading the state machine information from the state machine log mirroring mechanism and restoring the state of execution of a file transaction.

3. A file server performing file transaction operations in response to file transaction requests by the clients and including a state machine logging mechanism, comprising:

a storage sub-system, and

first and second control/processing sub-systems operating concurrently and in parallel, each including a file system processor performing file transaction operations in response to client requests directed to the first and second control/processing sub-systems and controlling file storage operations of the storage sub-system, and

a state machine logging mechanism operating concurrently and cooperatively with the file system processor, including a state machine log generator for extracting state

machine information defining at least one state machine during an execution of an operation, the at least one state machine representing a current state of execution of a file transaction of the corresponding control/processing sub-system wherein a state machine is comprised of state information including control and data values representing a state of operation of the control/processing sub-system at a given time, wherein

each control/processing sub-system is a state machine system defined during the execution of an operation by at least one sequential state machine defined by a state of operation of the state machine system during a step in the execution of the operation, and wherein

a file transaction operation is represented by at least state machine wherein each state machine is defined by data and control values residing in the state machine system during existence of state machine of the sequence, and

a state machine log for storing the state machine information of the corresponding control/processing sub-system, wherein the state machine log generator is responsive to the restoration of operation of the file server after a failure of the corresponding control/processing sub-system for reading the state machine information from the corresponding state machine log and restoring the state of execution of a file transaction of the corresponding control/processing sub-system.

4. The file server of claim 3, wherein each control/processing sub-system further comprises:

a state machine log mirroring mechanism that is functionally integral with the state machine log and that operates concurrently and in parallel with the state machine log and that is located separately from the state machine log mechanism and that is

communicating with the state machine log generator of the other control/processing sub-system for receiving and storing copies of the state machine information of the other control/processing sub-system, wherein the state machine log mirroring mechanism is responsive to the restoration of operation of the other control/processing sub-system after a failure of the other control/processing sub-system for reading the state machine information from the state machine log mirroring mechanism to the other control/processing sub-system and restoring the state of execution of a file transaction of the other control/processing sub-system. of execution of a file transaction of the other control/processing sub-system.

5. A system resource performing system resource operations in response to requests by the clients and including a state machine logging mechanism, comprising:

a system resource sub-system, and

a control/processing sub-system including a resource control processor performing system resource operations in response to client requests and controlling operations of the system resource sub-system, and

a state machine logging mechanism operating concurrently and cooperatively with the control/processing sub-system, including a state machine log generator for extracting state machine information defining at least one state machine during an execution of an operation, the at least one state machine representing a current state of execution of a system resource operation wherein a state machine is comprised of state information including control and data values representing a state of operation of the control/processing sub-system at a given time, wherein

the control/processing sub-system is a state machine system defined during the execution of an operation by at least one sequential state machine defined by a state of operation of the state machine system during a_step in the execution of the operation, and wherein

a resource operation is represented by at least one sequential state machine wherein each state machine is defined by data and control values residing in the state machine system during existence of state machine of the sequence, and

a state machine log for storing the state machine information, wherein the state machine log generator is responsive to the restoration of operation of the system resource after a failure of system resource operations for reading the state machine information from the state machine log and restoring the state of execution of a system resource operation.

6. The system resource of claim 5, wherein the state machine logging mechanism further comprises:

a state machine log mirroring mechanism that is functionally integral with the state machine log and that operates concurrently and in parallel with the state machine log and that is located separately from the control/processing sub-system and communicating with the state machine log generator for receiving and storing mirror copies of the state machine information, wherein the state machine log mirroring mechanism is responsive to the restoration of operation of the system resource after a failure of system resource operations for reading the mirror copies of the state machine information from the state machine log mirroring mechanism and restoring the state of execution of a system resource operation.

7. A system resource performing system resource operations in response to system resource requests by the clients and including a state machine logging mechanism, comprising:

a system resource sub-system, and

first and second control/processing sub-systems operating concurrently and in parallel, each including a system processor performing system resource operations in response to client requests directed to the first and second control/processing sub-systems and controlling operations of the system resource sub-system, and

a state machine logging mechanism operating concurrently and cooperatively with the control/processing sub-system, including a state machine log generator for extracting state machine information defining at least one state machine during an execution of an operation, the at least one state machine representing a current state of execution of a system resource operation of the corresponding control/processing sub-system wherein a state machine is comprised of state information including control and data values representing a state of operation of the control/processing sub-system at a given time, wherein

each control/processing sub-system is a state machine system defined during the execution of an operation by at least one sequential state machine defined by a state of operation of the state machine system during a step in the execution fo the operation, and wherein

a system resource operation is represented by at least one sequential state machine wherein each state machine is defined by data and control values residing in the state machine system during existence of state machine of the sequence, and

a state machine log for storing the state machine information of the corresponding control/processing sub-system, wherein the state machine log generator is responsive to the restoration of operation of the system resource after a failure of the corresponding control/processing sub-system for reading the state machine information from the corresponding state machine log and restoring the state of execution of a system resource operation of the corresponding control/processing sub-system.

8. The system resource of claim 7, wherein each control/processing sub-system further comprises:

a state machine log mirroring mechanism that is functionally integral with the state machine log and that concurrently and in parallel with the state machine log and that is located separately from the state machine log mechanism and that is communicating with the state machine log generator of the other control/processing sub-system for receiving and storing mirror copies of the state machine information of the other control/processing sub-system, wherein the state machine log mirroring mechanism is responsive to the restoration of operation of the other control/processing sub-system after a failure of the other control/processing sub-system for reading the mirror copies of the state machine information from the state machine log mirroring mechanism to the other control/processing sub-system and restoring the state of execution of a system resource operation of the other control/processing sub-system.

9. A state machine logging mechanism for use in a system resource performing system resource operations in response to requests by the clients, the system resource including a system resource sub-system and a control/processing sub-system including a resource control processor performing system resource operations in response to client

requests and controlling operations of the system resource sub-system, the state machine logging mechanism comprising:

a state machine log generator operating concurrently and cooperatively with the control/processing sub-system for extracting state machine information defining at least one state machine during an execution of an operation, the at least one state machine representing a current state of execution of a system resource operation wherein a state machine is comprised of state information including control and data values representing a state of operation of the control/processing sub-system at a given time, and

a state machine log for storing the state machine information, wherein the state machine log generator is responsive to the restoration of operation of the system resource after a failure of system resource operations for reading the state machine information from the state machine log and restoring the state of execution of a system resource operation wherein

the control/processing sub-system is a state machine system defined during the execution of an operation by at least one sequential state machine defined by state of operation of the state machine system during a step in the execution fo the operation, and wherein

a file transaction operation is represented by at least one sequential state machine wherein each state machine is defined by data and control values residing in the state machine system during existence of state machine of the sequence.

10. The state machine logging mechanism of claim 9, further comprising:

a state machine log mirroring mechanism that is functionally integral with the state machine log and operating concurrently and in parallel with the state machine log and

that is located separately from the control/processing sub-system and communicating with the state machine log generator for receiving and storing mirror copies of the state machine information, wherein the state machine log mirroring mechanism is responsive to the restoration of operation of the system resource after a failure of system resource operations for reading the mirror copies of the state machine information from the state machine log mirroring mechanism and restoring the state of execution of a system resource operation.

11. A state machine logging mechanism for use in a system resource performing system resource operations in response to system resource requests by the clients, the system resource including a system resource sub-system and first and second control/processing sub-systems, each including a system processor performing system resource operations in response to client requests directed to the first and second control/processing sub-systems and controlling operations of the system resource sub-system, the state machine logging mechanism comprising:

in each control/processor sub-system,

a state machine log generator for extracting state machine information defining at least one_state machine during an execution of an operation, the at least one state machine representing a state of execution of a system resource operation of the corresponding control/processing sub-system wherein a state machine is comprised of state information including control and data values representing a state of operation of the control/processing sub-system at a given time, and

a state machine log operating concurrently [[in]] <u>and</u> cooperatively with the corresponding control/processing sub-system for storing the state machine information of

the corresponding control/processing sub-system, wherein the state machine log generator is responsive to the restoration of operation of the system resource after a failure of the corresponding control/processing sub-system for reading the state machine information from the corresponding state machine log and restoring the state of execution of a system resource operation of the corresponding control/processing sub-system, wherein

each control/processing sub-system is a state machine system defined during the execution of an operation by at least one sequential state machine defined by a state of operation of the state machine system during a step in the execution fo the operation, and wherein

a file transaction operation is represented by at least one sequential state machine wherein each state machine is defined by data and control values residing in the state machine system during existence of state machine of the sequence.

12. The state machine logging mechanism of claim 11, further comprising:

in each control/processor sub-system,

a state machine log mirroring mechanism that is functionally integral with the state machine log and that operates concurrently and in parallel with the state machine log and that is located separately from the state machine log mechanism and that is communicating with the state machine log generator of the other control/processing sub-system for receiving and storing mirror copies of the state machine information of the other control/processing sub-system, wherein the state machine log mirroring mechanism is responsive to the restoration of operation of the other control/processing sub-system after a failure of the other control/processing sub-system for reading the mirror copies of the state machine information from the state machine log mirroring mechanism to the other

control/processing sub-system and restoring the state of execution of a system resource operation of the other control/processing sub-system.

13. In a system resource performing system resource operations in response to requests by the clients, the system resource including a system resource sub-system and a control/processing sub-system including a resource control processor performing system resource operations in response to client requests and controlling operations of the system resource sub-system and including a state machine logging mechanism, a method for logging and restoring the state of execution of system resource operations, comprising the steps of:

during each system resource operation,

extracting state machine information defining a sequence of state machines during an execution of an operation, each state machine representing a current state of execution of a system resource operation wherein a state machine is comprised of state information including control and data values representing a state of operation of the control/processing sub-system at a given time, and

storing the state machine information, and

upon restoration of operation of the system resource after a failure of system resource operations,

reading the state machine information from the state machine log and restoring the state of execution of a system resource operation, wherein

the control/processing sub-system is a state machine system defined during the execution of an operation by at least one sequential state machine defined by a state

of operation of the state machine system during a step in the execution fo the operation, and wherein

a file transaction operation is represented by at least one sequential state machine wherein each state machine is defined by data and control values residing in the state machine system during existence of state machine of the sequence.

14. The method for logging and restoring the state of execution of system resource operations of claim 13, further comprising the steps of:

during each system resource operation,

storing mirror copies of the state machine information separately from the control/processing sub-system, and

upon restoration of operation of the system resource after a failure of system resource operations,

reading the mirror copies of the state machine information and restoring the state of execution of a system resource operation.

15. In a system resource performing system resource operations in response to system resource requests by the clients, the system resource including a system resource sub-system and first and second control/processing sub-systems, each including a system processor performing system resource operations in response to client requests directed to the first and second control/processing sub-systems and controlling operations of the system resource sub-system, a method for logging and restoring the state of execution of system resource operations, comprising the steps of:

in each control/processor sub-system,

during each system resource operation,

extracting state machine information defining at least one state machine during an execution of an operation, the at least one state machine representing a current state of execution of a system resource operation of the corresponding control/processing sub-system wherein a state machine is comprised of state information including control and data values representing a state of operation of the control/processing sub-system at a given time, and

storing the state machine information of the corresponding control/processing sub-system, and

upon restoration of operation of the system resource after a failure of the corresponding control/processing sub-system,

reading the state machine information and restoring the state of execution of a system resource operation of the corresponding control/processing sub-system, wherein

each control/processing sub-system is a state machine system defined during the execution of an operation by at least one sequential state machine defined by a state of operation of the state machine system during a step in the execution fo the operation, and wherein

a file transaction operation is represented by at least one sequential state machine wherein each state machine is defined by data and control values residing in the state machine system during existence of state machine of the sequence.

16. The method for logging and restoring the state of execution of system resource operations of claim 15, further comprising the steps of:

in each control/processing sub-system,

during each system resource operation of the other control/processing sub-system,

receiving and storing mirror copies of the state machine information of the other control/processing sub-system, and

upon restoration of operation of the other control/processing sub-system after a failure of the other control/processing sub-system,

reading the state machine information from the state machine log mirroring mechanism to the other control/processing sub-system and restoring the state of execution of a system resource operation of the other control/processing sub-system.

APPENDIX B

PROPOSED AMENDED CLAIMS

1. A file server performing file transaction operations in response to file transaction requests by the clients and including a state machine logging mechanism, comprising:

a storage sub-system, and

a control/processing sub-system including

a file system processor performing file transaction operations in response to client requests and controlling file storage operations of the storage sub-system, and

a state machine logging mechanism operating concurrently and cooperative with the file system processor, including

a state machine log generator for extracting sequential state machine information defining ~~at least one~~a corresponding sequence of state machines during an execution of an operation, the ~~at least one~~sequence of state machines representing ~~a current~~sequential states of execution of a file transaction wherein a state machine is comprised of state information including control and data values representing a state of operation of the control/processing sub-system ~~at a given time~~during a corresponding step in the execution of the file transaction, wherein

the control/processing sub-system is a state machine system defined during the execution of a step in the execution of an operation by ~~at least one~~a sequential state machine defined by a state of operation of the state machine system during ~~a~~the step in the execution of the operation, and wherein

a file transaction operation is represented by ~~at least one~~ a sequence of ~~sequential~~ state machines wherein each state machine is defined by data and control values residing in the state machine system during existence of a state machine of the sequence, and

a state machine log for storing the state machine information, wherein the state machine log generator is responsive to the restoration of operation of the file server after a failure of file server operations for reading the state machine information from the state machine log and restoring the state of execution of a file transaction.

2. The file server of claim 1, wherein the state machine logging mechanism further comprises:

a state machine log mirroring mechanism that is functionally integral with the state machine log and that operates concurrently and in parallel with the state machine log and that is located separately from the control/processing sub-system and communicating with the state machine log generator for receiving and storing mirror copies of the state machine information, wherein the state machine log mirroring mechanism is responsive to the restoration of operation of the file server after a failure of file server operations for reading the state machine information from the state machine log mirroring mechanism and restoring the state of execution of a file transaction.

3. A file server performing file transaction operations in response to file transaction requests by the clients and including a state machine logging mechanism, comprising:

a storage sub-system, and

first and second control/processing sub-systems operating concurrently and in parallel, each including a file system processor performing file transaction operations in

response to client requests directed to the first and second control/processing sub-systems and controlling file storage operations of the storage sub-system, and

a state machine logging mechanism operating concurrently and cooperatively with the file system processor, including a state machine log generator for extracting state machine information defining ~~at least one~~ a sequence of state machines during an execution of an operation, ~~the at least one~~ each state machine representing a current state of execution of a step in the execution of a file transaction ~~of~~by the corresponding control/processing sub-system wherein a state machine is comprised of state information including control and data values representing a state of operation of the control/processing sub-system ~~at a given time~~during a corresponding step in the execution of a file transaction, wherein

each control/processing sub-system is a state machine system defined during the execution of an operation by ~~at least one sequential~~a sequence of state machines defined by a states of operation of the state machine system during a corresponding steps in the execution of the operation, and wherein

a file transaction operation is represented by at least ~~one sequential~~ a sequence of state machines wherein each state machine is defined by data and control values residing in the state machine system during existence of state machine of the sequence, and

a state machine log for storing the state machine information of the corresponding control/processing sub-system, wherein the state machine log generator is responsive to the restoration of operation of the file server after a failure of the corresponding control/processing sub-system for reading the state machine information

from the corresponding state machine log and restoring the state of execution of a file transaction of the corresponding control/processing sub-system.

4. The file server of claim 3, wherein each control/processing sub-system further comprises:

a state machine log mirroring mechanism that is functionally integral with the state machine log and that operates concurrently and in parallel with the state machine log and that is located separately from the state machine log mechanism and that is communicating with the state machine log generator of the other control/processing sub-system for receiving and storing copies of the state machine information of the other control/processing sub-system, wherein the state machine log mirroring mechanism is responsive to the restoration of operation of the other control/processing sub-system after a failure of the other control/processing sub-system for reading the state machine information from the state machine log mirroring mechanism to the other control/processing sub-system and restoring the state of execution of a file transaction of the other control/processing sub-system. of execution of a file transaction of the other control/processing sub-system.

5. A system resource performing system resource operations in response to requests by the clients and including a state machine logging mechanism, comprising:

a system resource-sub-system, and

a control/processing sub-system including a resource control processor performing system resource operations in response to client requests and controlling operations of the system resource sub-system, and

a state machine logging mechanism operating concurrently and cooperatively with the control/processing sub-system, including a state machine log generator for extracting state machine information defining ~~at least one~~a sequence of state machines during an execution of an operation, ~~the at least one~~each state machine representing a current state of execution of a corresponding step in the execution of a system resource operation wherein a state machine is comprised of state information including control and data values representing a state of operation of the control/processing sub-system ~~at a given time~~during a corresponding step in the execution in the system resource operation, wherein

the control/processing sub-system is a state machine system defined during the execution of an operation by ~~at least one sequential~~a sequence of state machines defined by ~~a~~ corresponding states of operation of the state machine system during ~~a~~ corresponding steps in the execution of the operation, and wherein

a resource operation is represented by ~~at least one sequential~~a sequence of state machines wherein each state machine is defined by data and control values residing in the state machine system during existence of a state machine of the sequence, and

a state machine log for storing the state machine information, wherein the state machine log generator is responsive to the restoration of operation of the system resource after a failure of system resource operations for reading the state machine information from the state machine log and restoring the state of execution of a system resource operation.

6. The system resource of claim 5, wherein the state machine logging mechanism further comprises:

a state machine log mirroring mechanism that is functionally integral with the state machine log and that operates concurrently and in parallel with the state machine log and that is located separately from the control/processing sub-system and communicating with the state machine log generator for receiving and storing mirror copies of the state machine information, wherein the state machine log mirroring mechanism is responsive to the restoration of operation of the system resource after a failure of system resource operations for reading the mirror copies of the state machine information from the state machine log mirroring mechanism and restoring the state of execution of a system resource operation.

7. A system resource performing system resource operations in response to system resource requests by the clients and including a state machine logging mechanism, comprising:

a system resource sub-system, and

first and second control/processing sub-systems operating concurrently and in parallel, each including a system processor performing system resource operations in response to client requests directed to the first and second control/processing sub-systems and controlling operations of the system resource sub-system, and

a state machine logging mechanism operating concurrently and cooperatively with the control/processing sub-system, including a state machine log generator for extracting state machine information defining ~~at least one~~ a sequence of state machines during an execution of an operation, ~~the at least one~~each state machine representing a ~~current~~ state of execution of a corresponding step in the exection of a system resource operation of the corresponding control/processing sub-system wherein a state machine is

comprised of state information including control and data values representing a state of operation of the control/processing sub-system ~~at a given time~~during the corresponding step in the execution of the system resource operation, wherein

each control/processing sub-system is a state machine system defined during the execution of an operation by ~~at least one sequential~~a sequence of state machines defined by a state of operation of the state machine system during a step in the execution ~~fo~~of the operation, and wherein

a system resource operation is represented by ~~at least one sequential~~ a sequence of state machines wherein each state machine is defined by data and control values residing in the state machine system during existence of a state machine of the sequence of state machines, and

a state machine log for storing the state machine information of the corresponding control/processing sub-system, wherein the state machine log generator is responsive to the restoration of operation of the system resource after a failure of the corresponding control/processing sub-system for reading the state machine information from the corresponding state machine log and restoring the state of execution of a system resource operation of the corresponding control/processing sub-system.

8. The system resource of claim 7, wherein each control/processing sub-system further comprises:

a state machine log mirroring mechanism that is functionally integral with the state machine log and that concurrently and in parallel with the state machine log and that is located separately from the state machine log mechanism and that is communicating with the state machine log generator of the other control/processing sub-system for

receiving and storing mirror copies of the state machine information of the other control/processing sub-system, wherein the state machine log mirroring mechanism is responsive to the restoration of operation of the other control/processing sub-system after a failure of the other control/processing sub-system for reading the mirror copies of the state machine information from the state machine log mirroring mechanism to the other control/processing sub-system and restoring the state of execution of a system resource operation of the other control/processing sub-system.

9. A state machine logging mechanism for use in a system resource performing system resource operations in response to requests by the clients, the system resource including a system resource sub-system and a control/processing sub-system including a resource control processor performing system resource operations in response to client requests and controlling operations of the system resource sub-system, the state machine logging mechanism comprising:

a state machine log generator operating concurrently and cooperatively with the control/processing sub-system for extracting state machine information defining ~~at least one~~a sequence of state machines during an execution of an operation, ~~the at least one~~each state machine representing a current state of execution of a step in the execution of a system resource operation wherein a state machine is comprised of state information including control and data values representing a state of operation of the control/processing sub-system ~~at a given time~~during execution of the corresponding step in the execution of the system resource operation, and

a state machine log for storing the state machine information, wherein the state machine log generator is responsive to the restoration of operation of the system

resource after a failure of system resource operations for reading the state machine information from the state machine log and restoring the state of execution of a system resource operation wherein

the control/processing sub-system is a state machine system defined during the execution of an operation by ~~at least one sequential~~a sequence of state machines defined by ~~a~~corresponding states of operation of the state machine system during a step in the execution ~~fo~~of the operation, and wherein

a file transaction operation is represented by ~~at least one sequential~~a sequence of state machines wherein each state machine is defined by data and control values residing in the state machine system during existence of the corresponding state machine of the sequence of state machines.

10. The state machine logging mechanism of claim 9, further comprising:

a state machine log mirroring mechanism that is functionally integral with the state machine log and operating concurrently and in parallel with the state machine log and that is located separately from the control/processing sub-system and communicating with the state machine log generator for receiving and storing mirror copies of the state machine information, wherein the state machine log mirroring mechanism is responsive to the restoration of operation of the system resource after a failure of system resource operations for reading the mirror copies of the state machine information from the state machine log mirroring mechanism and restoring the state of execution of a system resource operation.

11. A state machine logging mechanism for use in a system resource performing system resource operations in response to system resource requests by the clients, the

system resource including a system resource sub-system and first and second control/processing sub-systems, each including a system processor performing system resource operations in response to client requests directed to the first and second control/processing sub-systems and controlling operations of the system resource sub-system, the state machine logging mechanism comprising:

in each control/processor sub-system,

a state machine log generator for extracting state machine information defining ~~at least one~~a sequence of states machine during an execution of an operation, ~~the at least one~~each state machine representing a ~~current~~ state of execution of a step in the execution of a system resource operation of the corresponding control/processing sub-system wherein a state machine is comprised of state information including control and data values representing a state of operation of the control/processing sub-system ~~at a given time~~during execution of the corresponding step in the execution of an operation, and

a state machine log operating concurrently [[in]] and cooperatively with the corresponding control/processing sub-system for storing the state machine information of the corresponding control/processing sub-system, wherein the state machine log generator is responsive to the restoration of operation of the system resource after a failure of the corresponding control/processing sub-system for reading the state machine information from the corresponding state machine log and restoring the state of execution of a system resource operation of the corresponding control/processing sub-system, wherein

each control/processing sub-system is a state machine system defined during the execution of an operation by ~~at least one sequential~~a sequence of state machines

defined by a state of operation of the state machine system during a step in the execution ~~fo~~of the operation, and wherein

a file transaction operation is represented by ~~at least one sequential~~a sequence of states machine wherein each state machine is defined by data and control values residing in the state machine system during existence of the state machine of the sequence of state machines.

12. The state machine logging mechanism of claim 11, further comprising:

in each control/processor sub-system,

a state machine log mirroring mechanism that is functionally integral with the state machine log and that operates concurrently and in parallel with the state machine log and that is located separately from the state machine log mechanism and that is communicating with the state machine log generator of the other control/processing sub-system for receiving and storing mirror copies of the state machine information of the other control/processing sub-system, wherein the state machine log mirroring mechanism is responsive to the restoration of operation of the other control/processing sub-system after a failure of the other control/processing sub-system for reading the mirror copies of the state machine information from the state machine log mirroring mechanism to the other control/processing sub-system and restoring the state of execution of a system resource operation of the other control/processing sub-system.

13. In a system resource performing system resource operations in response to requests by the clients, the system resource including a system resource sub-system and a control/processing sub-system including a resource control processor performing system resource operations in response to client requests and controlling operations of the system

resource sub-system and including a state machine logging mechanism, a method for logging and restoring the state of execution of system resource operations, comprising the steps of:

during each system resource operation,

extracting state machine information defining ~~at least one~~a sequence of state machines during an execution of an operation, ~~the at least one~~each state machine representing a current state of execution of a corresponding step in an execution of a system resource operation wherein a state machine is comprised of state information including control and data values representing a state of operation of the control/processing sub-system ~~at a given time~~during the corresponding step in the execution of the system resource operation, and

storing the state machine information, and

upon restoration of operation of the system resource after a failure of system resource operations,

reading the state machine information from the state machine log and restoring the state of execution of a system resource operation, wherein

the control/processing sub-system is a state machine system defined during the execution of an operation by ~~at least one sequential~~a sequence of state machines defined by ~~a~~ states of operation of the state machine system during ~~a~~corresponding steps in the execution ~~fo~~of the operation, and wherein

a file transaction operation is represented by at least one sequential state machine wherein each state machine is defined by data and control values residing in the state machine system during existence of state machine of the sequence.

14. The method for logging and restoring the state of execution of system resource operations of claim 13, further comprising the steps of:

during each system resource operation,

storing mirror copies of the state machine information separately from the control/processing sub-system, and

upon restoration of operation of the system resource after a failure of system resource operations,

reading the mirror copies of the state machine information and restoring the state of execution of a system resource operation.

15. In a system resource performing system resource operations in response to system resource requests by the clients, the system resource including a system resource sub-system and first and second control/processing sub-systems, each including a system processor performing system resource operations in response to client requests directed to the first and second control/processing sub-systems and controlling operations of the system resource sub-system, a method for logging and restoring the state of execution of system resource operations, comprising the steps of:

in each control/processor sub-system,

during each system resource operation,

extracting state machine information defining ~~at least one~~a sequence of state machines during an execution of an operation, ~~the at least one~~each state machine representing a current state of execution of a step in an execution of a system resource operation of the corresponding control/processing sub-system wherein a state machine is comprised of state information including control and data values representing a state of

operation of the control/processing sub-system ~~at a given time~~during a corresponding step in the execution of the system resource operation, and

storing the state machine information of the corresponding control/processing sub-system, and

upon restoration of operation of the system resource after a failure of the corresponding control/processing sub-system,

reading the state machine information and restoring the state of execution of a system resource operation of the corresponding control/processing sub-system, wherein

each control/processing sub-system is a state machine system defined during the execution of an operation by ~~at least one sequential~~a sequence of state machines defined by ~~a~~corresponding states of operation of the state machine system during ~~a~~corresponding steps in the execution ~~to~~of the operation, and wherein

a file transaction operation is represented by ~~at least one sequential~~a sequence of state machines wherein each state machine is defined by data and control values residing in the state machine system during existence of a state machine of the sequence of state machines.

16. The method for logging and restoring the state of execution of system resource operations of claim 15, further comprising the steps of:

in each control/processing sub-system,

during each system resource operation of the other control/processing sub-system,

receiving and storing mirror copies of the state machine information of the other control/processing sub-system, and

upon restoration of operation of the other control/processing sub-system after a failure of the other control/processing sub-system,

reading the state machine information from the state machine log mirroring mechanism to the other control/processing sub-system and restoring the state of execution of a system resource operation of the other control/processing sub-system.